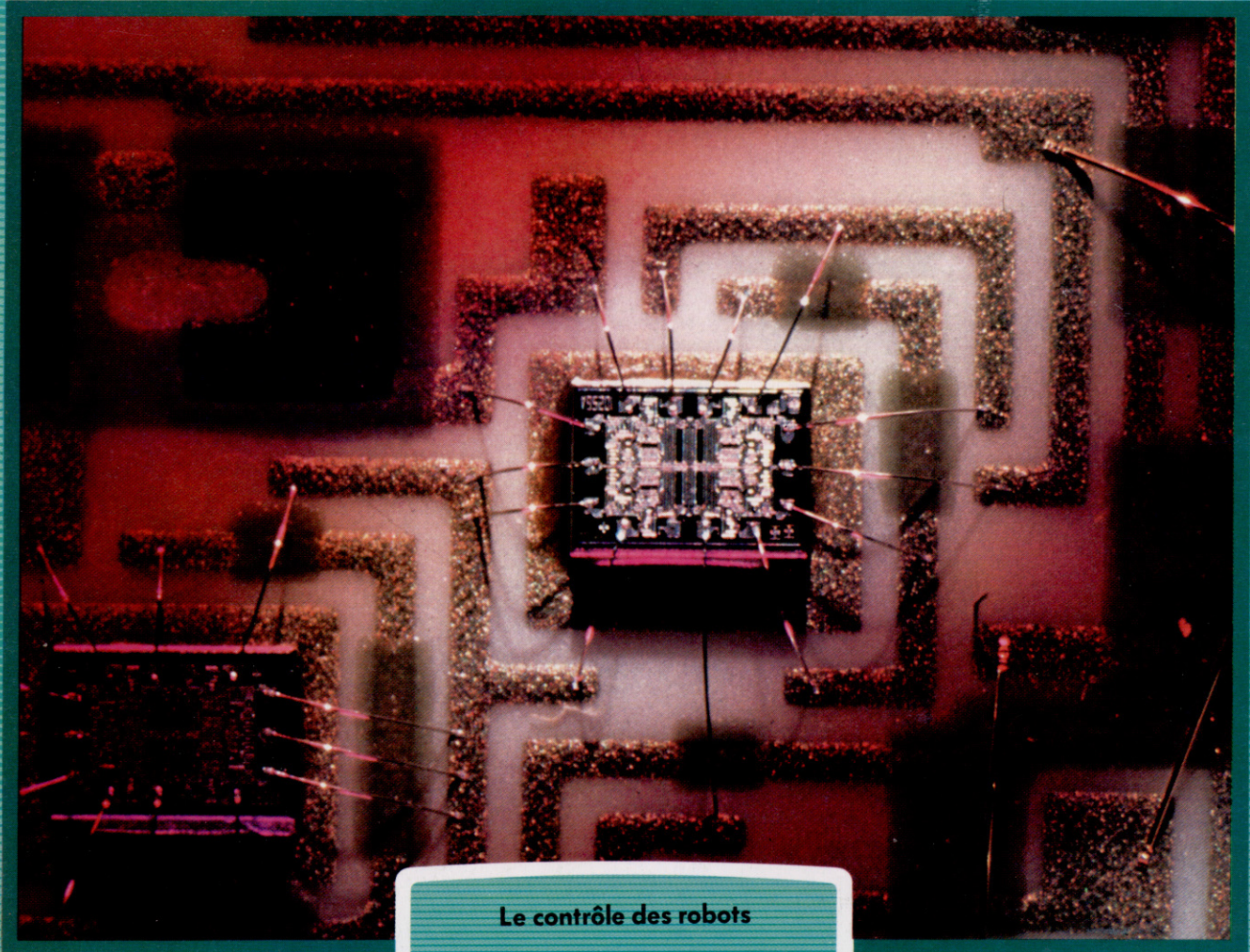


# abc

N° 57

COURS  
D'INFORMATIQUE  
PRATIQUE  
ET FAMILIALE

# INFORMATIQUE



Le contrôle des robots

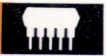
Le micro gère la maison

Le Tandy Modèle 100

La carte à puce

EDITIONS  
**ATLAS**

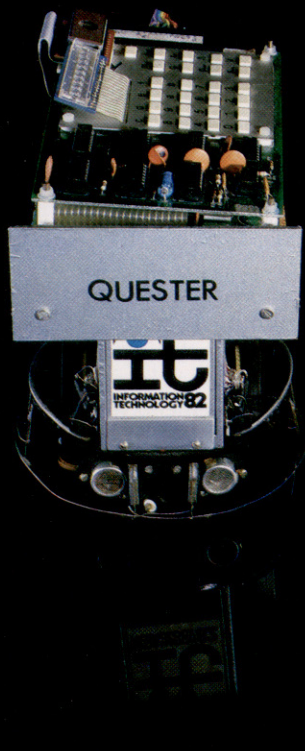
**Page manquante  
(publicité et colophon)**



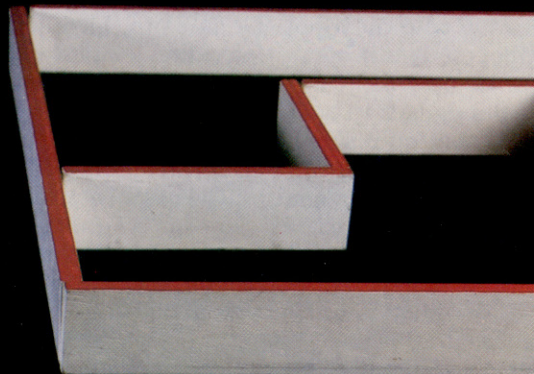
# La voie étroite

Une fois animé, de préférence par des moteurs électriques, le robot doit être dirigé pour aller là où on le désire. Nous étudions ici le contrôle du déplacement des robots.

## La souris et le labyrinthe



**La souris...**  
Quester, montré ici, représente une recherche en robotique avec une souris-robot parcourant un labyrinthe. Pour de nombreux amateurs de robotique, Quester a permis l'acquisition de connaissances pratiques et techniques. Le créateur du robot a mis en œuvre un ensemble complet de capteurs optiques, acoustiques et sensitifs.



**et le labyrinthe**  
Les micro-souris disposent d'une période d'expérimentation pour apprendre à connaître le labyrinthe par toutes méthodes autres que celles faisant intervenir la communication avec l'extérieur. Elles doivent ensuite, en une épreuve contre la montre, tenter d'en gagner le centre.

Marcus Willy

La méthode la plus simple pour animer un robot suppose qu'il possède une carte spéciale lue par un mécanisme. Le contour de la carte est suivi par un petit arbre à came qui influe à son tour sur une suite de leviers contrôlant la direction. Il y a peu de temps encore, on pouvait acheter des modèles réduits de voitures et de petits robots-jouets qui fonctionnaient de la sorte. Leur programme se faisait en découpant une carte à l'aide de ciseaux. Le robot se déplaçait selon les dentelures de cette dernière. D'autres robots utilisaient des dispositifs leur permettant de suivre, au moyen de relais électromécaniques, une route prédéterminée. Ces méthodes mécaniques étaient toutefois limitées pour la simple raison que les différents éléments mécaniques en question devenaient trop chers et relativement inefficaces. Elles permirent, cependant, de véritablement essayer les plâtres.

Ce procédé est semblable à celui utilisé par les voitures de course en modèle réduit, qui dispo-

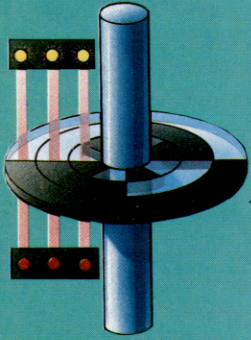
saient d'une sorte de balai-guide se déplaçant à l'intérieur d'une fente où circule l'électricité. Les deux formes de robots « téléguidés » les plus connues sont celles qui, d'une part, suivent une ligne tracée sur le sol et celles qui, d'autre part, sont « filoguidées » (reliées à un fil).

De nos jours, une des meilleures méthodes consiste à tracer sur le sol le parcours que le robot devra suivre. Les robots qui suivent une ligne tracée sur le sol le font grâce à un capteur, par exemple une cellule photoélectrique ou un capteur à infrarouges, qui informe le robot s'il est dans une zone de lumière ou d'ombre. Si, par exemple, la couleur de fond est foncée et si la ligne du parcours est de couleur claire, le signal en sortie du capteur sera toujours le plus élevé lorsque le robot est sur la ligne.

Cette technique pose toutefois un problème : que fait le robot lorsque le signal émis par le senseur chute, indiquant qu'il est sorti de la ligne ? Avec un système à un seul capteur, le mieux que



### Axe encodeur



Ce dispositif permet au robot d'apprécier son déplacement selon la rotation de ses axes. Il consiste en un disque calibré monté sur un axe. Le disque est divisé en un certain nombre de cercles concentriques étalonnés en zones transparentes ou opaques. Avec une source lumineuse et un capteur lumineux pour chaque cercle, il est possible de déterminer l'orientation précise de l'axe.

La précision d'un axe-encodeur dépend de son nombre d'anneaux. L'illustration montre un axe avec trois anneaux pouvant encoder les nombres binaires compris entre 000 et 111 (de 1 à 7 en décimal). La précision est ici de  $360/8 = 45^\circ$ . Un axe de huit anneaux donnerait une précision de  $45^\circ$ .

le robot puisse faire est d'errer jusqu'à ce que le signal remonte, lui indiquant qu'il est à nouveau sur la ligne. Il peut alors continuer dans sa direction. Ce système n'est pas aussi aléatoire qu'il paraît. Ainsi, si le robot allait vers la gauche lorsque le signal a faibli, il est évident qu'il tournera vers la droite pour retrouver l'émission. De même, ayant retrouvé la ligne, il supposera que le chemin à suivre se trouve quelque part entre la route, mauvaise, sur la gauche, qu'il suivait lorsqu'il s'est égaré et celle vers la droite qui l'a remis dans le bon chemin.

Afin de réduire le temps nécessaire à un robot « déraillant » pour se remettre sur la ligne, le système devra comprendre deux capteurs, un de chaque côté du robot. De la sorte, lorsque le robot est dans la bonne trajectoire, le signal émis par les deux capteurs est faible; s'il s'en écarte, un des signaux va croître. Le robot sait alors non seulement qu'il s'écarte, mais aussi dans quelle direction il s'écarte du bon chemin : lorsqu'il va vers la droite, le signal gauche augmente, puisque le capteur gauche est alors sur la ligne claire, et inversement. Le robot comprend cette modification comme une instruction, et il y répondra en s'orientant dans la direction du capteur dont le signal est plus puissant.

Ce système peut aussi bien fonctionner avec une ligne foncée sur un fond clair, l'important étant la différence d'intensité lumineuse entre le fond et la ligne, ainsi que la programmation de la réaction du robot.

L'autre système de guidage des robots consiste à placer dans le sol un fil électrique. Le faible courant émis génère un champ magnétique autour du fil. Ce champ est détecté par un capteur assez simple — une petite bobine relèvera

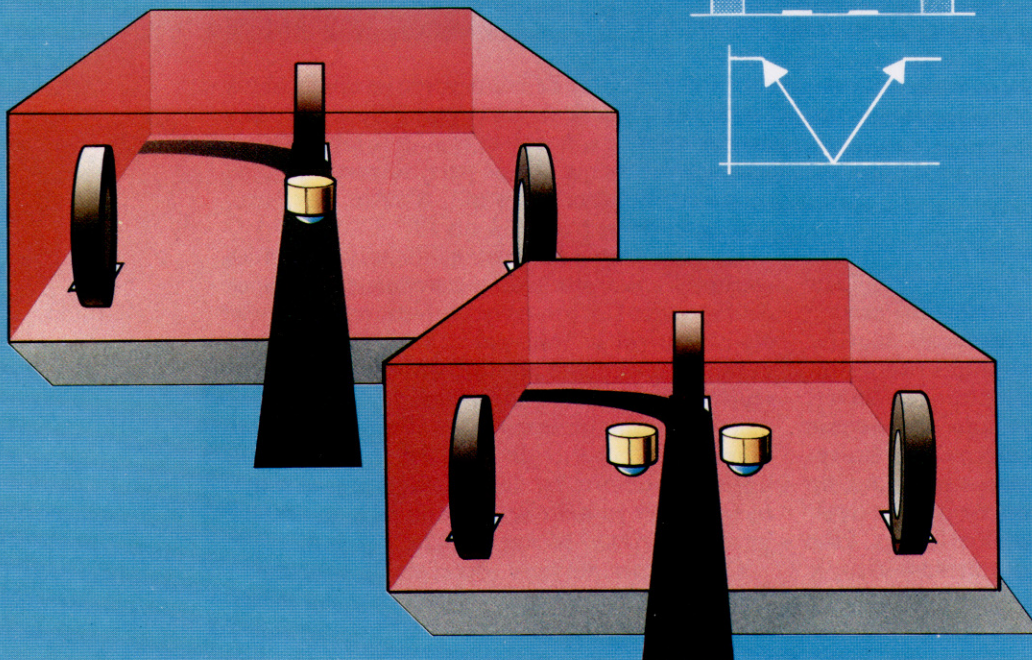
le champ émis et produira un faible voltage, susceptible d'être amplifié pour agir à la manière du signal lié à une intensité lumineuse. Les robots utilisés dans l'industrie dépendent souvent, pour leurs déplacements, d'un fil électrique enfoui dans le sol. Ce système s'est révélé plus sûr que l'autre, dont le bon fonctionnement dépend de l'état de propreté du sol : la ligne peinte doit rester propre, et le contraste avec le fond bien marqué.

### Télécommande

Une autre méthode consiste à contrôler le déplacement du robot par télécommande, ce qui est particulièrement intéressant lorsqu'il s'agit de tâches présentant des dangers pour l'homme, des tâches telles que le maniement des bombes, de matières radioactives ou chimiques, mais aussi tout travail dans des conditions défavorables (chaleur, froid, humidité), voire nuisibles à l'homme. Un engin célèbre de ce type est le robot soviétique Lounokhod qui a été déposé sur la Lune par Luna 16 en 1970. Il s'agissait d'un robot sur roues qui devait récolter des informations sur la surface lunaire, sous contrôle radio d'opérateurs sur Terre.

Le contrôle des robots de ce type est peu différent du contrôle d'un modèle réduit d'avion télécommandé. Le signal radio peut être soit un signal analogique qui varie en intensité selon l'importance des mouvements à effectuer, soit un signal numérique qui constitue une configuration de bits donnant des détails sur les mouvements à suivre. Les communications analogiques semblent moins heureuses que les communications numériques, car des facteurs étrangers peuvent

### Capteurs lumineux



Les robots peuvent suivre des traces sur le sol au moyen de capteurs de lumière. La trace peut être de couleur claire sur un fond sombre, ou inversement. Dans l'un et l'autre cas, une cellule doit détecter une modification de luminosité sur le sol.

Avec un seul capteur de piste, le robot ne peut rien savoir d'autre que s'il est bien sur la ligne ou s'il ne l'est pas. S'il s'en écarte, il ne lui reste qu'à « tâtonner » pour retrouver son chemin. Avec un système double piste utilisant une piste sombre et deux capteurs de part et d'autre, le robot sait qu'il est en bonne voie lorsque les deux capteurs détectent le fond clair. Lorsque le robot s'écarte du chemin, l'un des capteurs transmet un signal plus élevé.

(Cl. Kevin Jones.)



interférer avec l'intensité du signal de la transmission analogique. Si vous essayez de recevoir un signal radio lointain, vous remarquerez combien la réception est inégale selon l'heure et les conditions climatiques. Le même genre de problème peut affecter les communications radio pour la télécommande de robots.

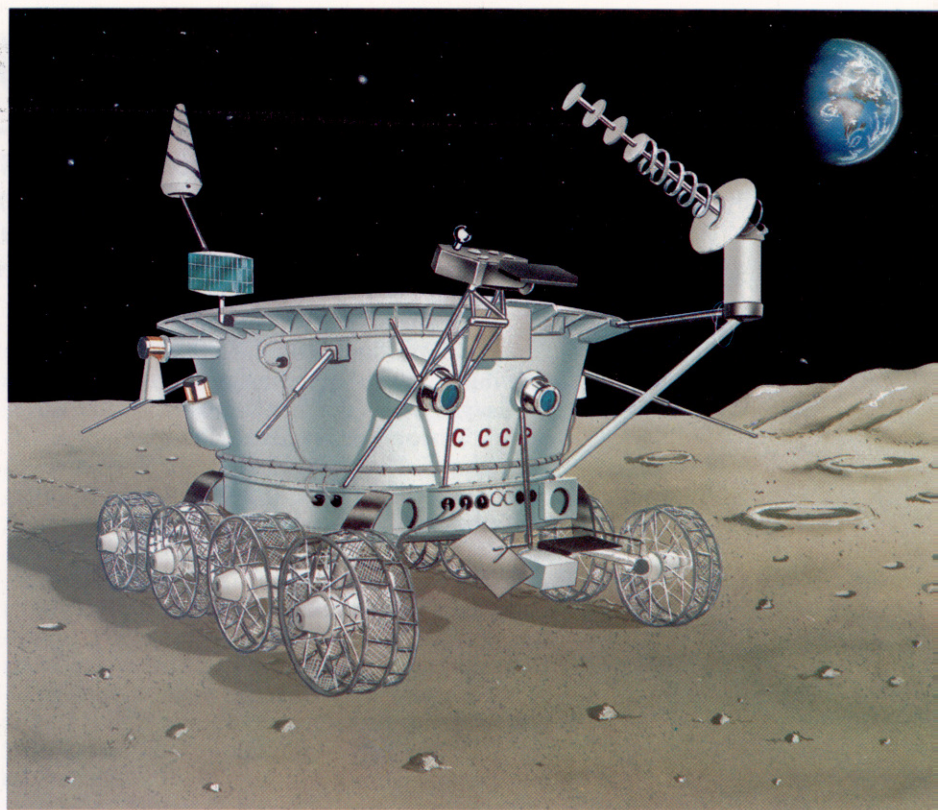
Les méthodes numériques ne sont pas exemptes de problèmes ; ils paraissent surtout lorsque des interférences suppriment des bits ou en insèrent de manière intempestive. Afin de les éviter, les messages sont souvent répétés ; le robot n'y obéit que lorsqu'il a reçu plusieurs fois un message identique.

## Systemes rétroactifs

Une technique plus sophistiquée consiste à mettre en œuvre un système de boucle, lequel permet au robot d'envoyer un signal de retour (rétroaction) à l'émetteur, concernant le signal qu'il vient de recevoir. Il s'agit en quelque sorte d'un dialogue entre l'émetteur et le robot. Par exemple, l'émetteur dit au robot « avance », le robot ayant reçu le message demande confirmation : « Avez-vous dit "avance" ? » ; l'émetteur confirme, et c'est seulement alors que le robot s'exécute. Cela peut éviter de sérieuses erreurs, par exemple lorsque le robot manipule des déchets radioactifs ou qu'il doit descendre dans un cratère lunaire.

Les mêmes principes peuvent être appliqués à la télécommande. Par exemple, certains robots peuvent être contrôlés au moyen d'émetteurs à infrarouges, du type de ceux utilisés pour la télécommande de téléviseurs. Les robots pourraient également être contrôlés par infra-sons ou encore par des sons audibles mais de nature spécifique, par exemple des suites de battements de mains. Quelle que soit la technique retenue, les méthodes sous-jacentes de communication et de vérification des informations restent les mêmes. Si l'opérateur est à proximité du robot, ces techniques ne sont pas absolument nécessaires. Les commandes peuvent alors être transmises au robot par câble de liaison. Il est également possible d'utiliser plusieurs câbles, ce qui revient à utiliser plusieurs canaux pour un avion télécommandé par radio. La différence, dans le cas du robot, apparaît dans le câble supplémentaire qui est utilisé pour une communication en parallèle et non pas en série (une chaîne de bits en parallèle sur les deux câbles et non une suite d'impulsions sur un seul câble). Cela permet une communication plus rapide ; en outre, la plupart des ordinateurs comportent un port parallèle. Il est donc possible de transmettre les instructions par l'intermédiaire du clavier d'un ordinateur.

Si les mouvements du robot sont contrôlés par un opérateur au clavier d'un ordinateur, l'utilisation de l'ordinateur ne complique pas la conception du robot : ce qui compte, c'est la présence de l'homme et le contrôle visuel du robot. En revanche, si le robot est éloigné (sur la Lune, ou tout simplement dans une pièce voisine) ou s'il doit être contrôlé par un programme et non



plus en temps réel, le robot doit alors être sensiblement plus intelligent.

L'important est d'obtenir une certaine forme de rétroaction. Il s'agit d'un procédé qui permet au système d'adapter ce qu'il fait en fonction de ce qu'il vient de faire et de ce qu'il doit faire. Ainsi, si vous voulez que le robot avance de trois pas, et si vous le contrôlez directement, vous pouvez le mettre en marche, apprécier son avance et l'arrêter lorsqu'il a fait trois pas. Vous le pouvez parce que vous disposez du retour de l'information transmise. Vous agissez selon cette réponse et vous corrigez le cours des choses.

Les sens humains, surtout la vue en l'occurrence, constituent ici ce retour d'informations. En l'absence de ce dernier, le robot doit savoir précisément tout ce qu'il a à faire. Un robot qui se guide sur une ligne tracée sur le sol en reçoit les informations qui lui permettent d'agir. De même, un robot piloté par ordinateur doit utiliser une certaine forme de rétroaction pour avancer, par exemple exactement de trois pas. Une des méthodes les plus couramment employées a recours à un axe-encodeur. Il s'agit d'un disque relié aux transmissions principales du robot et qui donne une mesure très précise de leurs mouvements. Aussi, lorsque l'ordinateur demande au robot d'avancer de trois pas, le robot se met en marche et, simultanément, prend connaissance du signal en provenance des axes-encodeurs qui lui indiquent sa position. Le robot peut donc adapter son action : continuer d'avancer, s'arrêter s'il a atteint le terme, revenir en arrière s'il le faut. Dans ce dernier cas, il revient sur ses pas d'une distance égale à ce qu'il a calculé d'après les indications des axes-encodeurs.

**Un pas de géant pour l'espèce robot**  
Lounokhod 1, robot soviétique, fut déposé sur la Lune en 1970 pour réunir des informations sur la nature du sol et de l'atmosphère lunaires. Ce n'était pas un vrai robot — étant contrôlé depuis la Terre —, mais il permit de rapporter une moisson d'échantillons scientifiques. Comme tous les engins télécommandés lancés dans l'espace, Lounokhod a été handicapé par le temps de latence (décalage), de trois secondes, des transmissions entre la Terre et la Lune, entre le moment où l'information est émise vers la Terre et celui où il reçoit en retour la réponse, le signal de contrôle. (Cl. Steve Cross.)



# Grands ensembles

Des logiciels comme Lotus 1-2-3, Symphony ou Xchange de Psion sont des programmes destinés à la gestion de haut niveau. Mais leurs techniques seront bientôt d'usage général.

Comme nous l'avons déjà vu, un logiciel intégré doit faire partie d'un environnement qui permette à l'utilisateur d'accéder instantanément aux différentes tâches qui se révéleront indispensables. Les procédures d'exploitation resteront les mêmes quel que soit le type d'application envisagé; l'information doit pouvoir circuler librement entre les applications.

Lotus 1-2-3 reprend le format, déjà classique, du tableur : chiffres et formules sont placés dans une matrice de « cases » et peuvent être modifiés sans effort, puis calculés de nouveau. Mais ce programme offre bien des possibilités supplémentaires, qui en font autre chose qu'un simple outil d'analyse et de prévision financières. Il est par exemple tout à fait possible d'y stocker des informations très diverses : noms, adresses, numéros de téléphone, et aussi données numériques. Une zone déterminée de la grille peut servir de table de référence, et elle contiendra tous les détails utiles — disons une liste de clients, avec le numéro de compte de chacun. Lotus 1-2-3 rend très simple le traitement et la réorganisation de tous ces renseignements : c'est dire que le tout

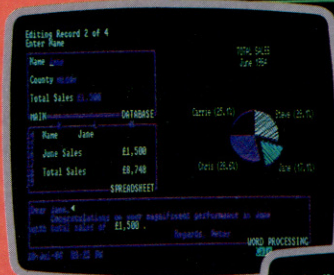
joue le rôle d'une petite base de données. Il est, de même, parfaitement possible de n'en extraire que les cases comportant des données numériques, puis de les représenter à l'écran par un ensemble de graphiques, ce qui rend inutile l'existence d'un programme spécialisé. Lotus 1-2-3 est de surcroît capable de manipuler du texte, et l'on peut donc en faire usage pour rédiger des notes de service ou de brèves analyses, bien que l'espace mémoire, toujours limité, empêche d'y voir un véritable traitement de texte.

Le regroupement d'options aussi diversifiées fait de Lotus 1-2-3 le seul programme dont bien des usagers auront jamais besoin. Une seule feuille de calcul accueille les informations nécessaires à des applications différentes : il est donc très facile de parvenir à des résultats inconcevables avec des logiciels conventionnels. Imaginons, par exemple, qu'un commerçant possède plusieurs kiosques à journaux dans une grande ville et qu'il veuille disposer d'un relevé précis, hebdomadaire, mensuel, trimestriel et annuel, des ventes de chacun d'eux. Le meilleur moyen pour ce faire consiste à regrouper tous les kiosques,

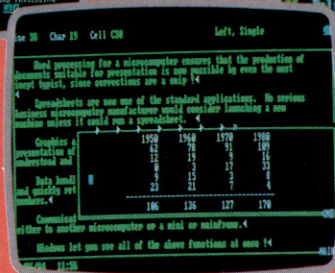
**Variations symphoniques**  
Symphony est un logiciel intégré parce qu'il transforme la mémoire en une énorme feuille de travail. Elle permet l'accès aux données par l'intermédiaire de plusieurs fenêtres, dont chacune accomplit une fonction différente : traitement de texte, base de données, tableur ou affichage graphique. Cela résout le problème des échanges d'informations mais exige de très grosses quantités de RAM.

Base de données

Name	1950	1960	1970	1980
Address 1	62	78	91	109
Address 2	12	19	9	16
Address 3	3	17	33	33
Telephone	9	15	3	8
	23	21	7	4



Affichage graphique



Traitement de texte

## Symphony

Feuille de travail principale

	1950	1960	1970	1980
Income Tax	62	78	91	109
Corporation Tax	12	19	9	16
VAT	3	17	33	33
Customs Duty	9	15	3	8
Other	23	21	7	4
	106	136	127	170
Defence	9	14	15	26
Health Care	26	37	36	31
Education	19	29	33	38
Social Services	8	16	27	30
Police	17	16	17	25
Trucks & Rail	18	9	5	18
	97	121	133	160
Surplus/Deficit	9	15	-6	10



ainsi que leurs chiffres d'affaires, au sein d'une unique feuille de calcul. Les formules nécessaires sont rédigées de telle façon que les seuls chiffres à modifier soient les recettes hebdomadaires de tous les points de vente; les autres résultats sont alors ajustés automatiquement.

C'est d'ailleurs ce qu'un tableur classique fait couramment. Mais que se passera-t-il si le commerçant veut classer chaque emplacement en fonction de ses ventes, le plus important se retrouvant en tête de liste? Il faudra d'abord les entrer en suivant l'ordre alphabétique, puis effectuer un tri, qu'il sera indispensable de renouveler toutes les semaines. Lotus 1-2-3 accomplit ce type de travail très vite et très facilement. Le commerçant peut désirer connaître, par l'intermédiaire d'un tableau hebdomadaire, comment chaque emplacement s'est comporté commercialement. Cette demande ne nécessite que quelques manipulations au clavier pour obtenir un résultat par le tableur, un affichage sur l'écran et une copie papier.

## Symphony

Le nouveau produit de Lotus, Symphony, reprend la même idée de base : l'emploi d'un tableur pour stocker des données utilisables par plusieurs programmes d'application différents. Mais Symphony permet de surcroît à l'utilisateur de diviser l'écran en « fenêtres », dont chacune est consacrée à une partie spéciale de la feuille de calcul et formatée en fonction des informations qu'elle affiche.

Si, par exemple, il s'agit de texte, on aura un de ces petits écrans comme en ont les traitements de texte, qui indiquent les marges et les tabulations. S'il s'agit d'un graphique, les axes en seront libellés et échelonnés. Pour une base de données, chaque entrée aura droit à son propre écran, un peu comme dans un système de fiches cartonnées. Symphony donne donc l'impression de manipuler simultanément quatre programmes d'applications séparés; mais c'est en fait un tableur surpuissant.

Tout comme Lotus 1-2-3, il peut d'ailleurs « apprendre » à reconnaître des séquences de touches au clavier : l'utilisateur peut ainsi rendre presque automatiques certaines opérations fréquemment renouvelées (on appelle macro-programmes les routines qui gèrent cet aspect particulier). Symphony est également équipé de son propre langage de programmation évolué, dans lequel on peut au besoin rédiger telle ou telle application (facturation, contrôle de stock). Il est même plus facile de procéder de cette façon que de recourir au BASIC, puisque Symphony se charge du traitement graphique ou de la recherche et de l'organisation des données. Les logiciels spécifiques sont eux aussi stockés sur la feuille de travail et peuvent donc accéder à toutes les options disponibles.

Si puissant qu'il soit, ce logiciel ne manque pas de concurrents. L'un des plus redoutables est Framework, d'Ashton Tate, dont le fonctionnement est très semblable; mais sa structure de données

## Xchange

Programmes d'application

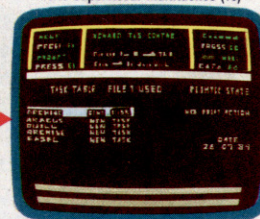
Abacus - Tableur

Archive - Base de données

Easel - Affichage graphique

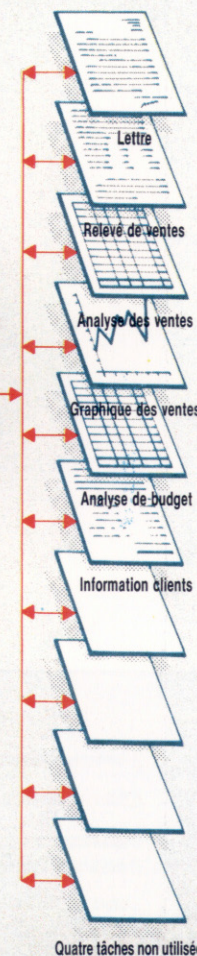
Quill - Traitement de texte

Menu superviseur multitâches (10)



Dû à la firme britannique Psion, Xchange regroupe plusieurs programmes (tableur, base de données, traitement de texte, affichage graphique), dont chacun est stocké sur disquette, accompagné par un logiciel qui sert de superviseur pouvant gérer jusqu'à dix programmes à la fois. Abandonner une « tâche » permet d'accéder — par l'intermédiaire du menu — à toutes les autres : le superviseur chargera et exécutera toute application qui lui sera demandée. Les commandes EXPORT et IMPORT permettent l'échange des données d'une « tâche » à l'autre;

elles créent et gèrent des fichiers sur disquettes correspondant aux données particulières.



est encore plus complexe et dissimulée. Le gros inconvénient de chacun d'eux, outre leur prix, est une importante consommation d'espace mémoire. Symphony exige au moins 320 K de RAM mais ne donne vraiment toute sa mesure qu'avec 512 K. Il s'en suit que seuls des ordinateurs 16 bits sont capables de les accueillir. Pourtant, ils n'imposent pas un échange d'informations périodique entre disquettes et mémoire principale, comme c'est le cas pour la plupart des programmes de gestion.

En théorie, bien sûr, le coût des mémoires devrait baisser sans cesse, ce qui les mettrait toujours plus à la portée de l'utilisateur moyen. En pratique pourtant, il n'en va pas ainsi, et il faudra sans doute attendre encore un certain temps avant que les choses changent. Un logiciel comme Symphony, s'il crée véritablement de nouvelles normes, reste limité par le matériel sur lequel il tourne et ne peut remplir sa fonction que parce qu'il a été très soigneusement conçu.

Une autre méthode d'intégration a été mise au point au cours de ces vingt dernières années, et certains programmes qui la mettent en œuvre ont déjà fait leur apparition sur le marché; nous passerons en revue les perspectives qui s'offrent à ce type de logiciel.



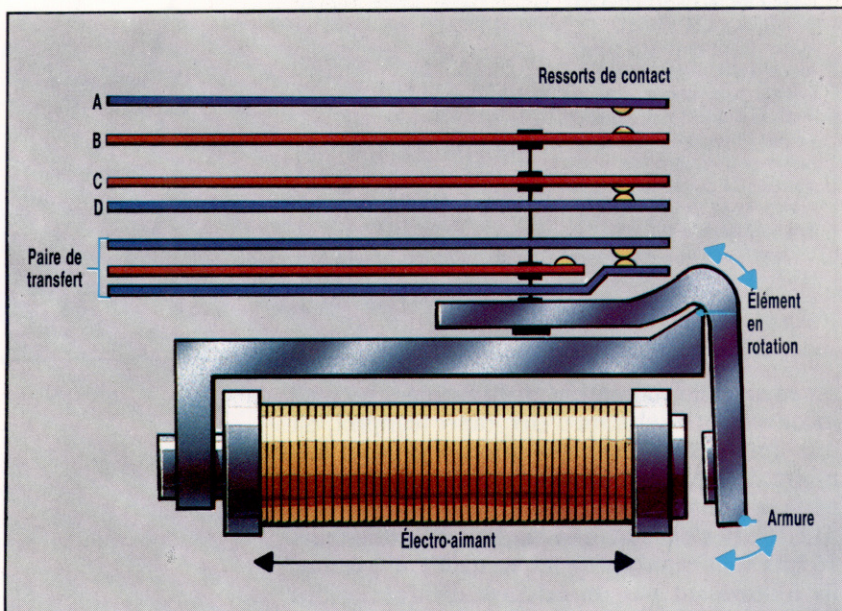
# Bonne alimentation

Grâce à un boîtier relais, votre ordinateur sera capable, entre autres, d'éteindre et d'allumer les lumières de la maison à des intervalles prédéterminés ou aléatoires.

Les relais électriques sont des interrupteurs pouvant être sollicités par un signal électrique. Dans notre application, les relais servent à commander des appareils à haute tension et à haute intensité en utilisant un signal de basse tension et de faible intensité. Il existe de nombreux types de relais, mais le plus commun est le type à armature, qui utilise un solénoïde pour établir ou couper les connexions.

Le relais établit et coupe le contact grâce à de petits déplacements de l'armature. Une tension appropriée appliquée à la bobine du solénoïde génère un champ magnétique, qui attire l'arma-

ture. Lorsque le solénoïde est actif, le ressort du milieu monte et établit un contact avec le ressort supérieur, ce qui brise le contact entre les ressorts du haut et du bas.



ture. Lorsque cela se produit, les contacts à ressort situés à l'autre extrémité sont poussés verticalement vers le haut.

L'illustration ci-dessus montre la position non active : aucune tension n'est appliquée au solénoïde. Dans cette position, la paire de contacts AB est ouverte, et la paire CD est fermée. Lorsque le solénoïde est actif, les ressorts B et C se déplacent vers le haut, ce qui fait fermer A et B, et ouvrir C et D. Cet arrangement peut être utilisé de deux façons : soit pour fermer un circuit tout en ouvrant un autre circuit, soit plus simplement, pour compléter ou pour rompre un circuit.

Un relais peut aussi servir de mécanisme de transfert. Dans le schéma, les trois ressorts inférieurs sont disposés de façon telle que — dans la position non active — les ressorts du haut et

## Liste des pièces

Quantité	Article	N° Maplin
2	Relais 10 A 240 V contact 8-15 V	YX97F
2	Prise de secteur non commuté	HL68Y
2	Boîtier 29 mm	YB15R
2 m	Câble secteur 6 A 3 voies	XR03D
2	Fiche secteur, fusible 13 A	RW67X
*	Fiche 4 mm	*
*	0,5 m de câble-ruban 2 voies	*
*	Petits morceaux de carte de montage	*

**N.B. :** Les articles marqués d'un astérisque résultent des projets précédents. Ces quantités correspondent à deux boîtiers relais à une prise. Le boîtier de sortie peut piloter quatre boîtiers de ce type — si on le désire, ceux-ci peuvent être des boîtiers à prises doubles, voire triples ; les principes de construction demeurent les mêmes, seules les pièces diffèrent.

## Intérieur du boîtier

Vérifiez la sécurité et la conformité de toutes les connexions, dont l'absence de shuntage entre les pistes du montage. Assurez-vous qu'il n'y a aucune connexion électrique entre le conducteur d'alimentation et les lignes de signaux.

Collez la carte sur un coin du boîtier en utilisant une colle époxy (évités d'utiliser les colles domestiques conductrices d'électricité). Si vous doutez de la conductivité de votre adhésif, étendez une mince couche de colle sur un morceau de carte, laissez sécher, puis connectez un multimètre à chaque extrémité : si le compteur donne une lecture positive, changez de colle !

Dès que la colle est sèche, vissez le couvercle sur le boîtier et mettez les quatre fiches 4 mm sur les lignes de signaux (celles-ci peuvent être de la même couleur, puisque le relais fonctionnera quelle que soit la direction du courant). Connectez ensuite la fiche 13 A au conducteur de secteur. Le relais est étalonné à 10 A, mais, pour plus de sécurité, vous ne devriez pas lui soumettre plus de 5 A ; mettez donc un fusible de 5 A dans la fiche : cela vous permet de contrôler des appareils d'une puissance allant jusqu'à 1,2 kW.





## Avertissement!

Ceci est un projet très simple, mais tout ce qui touche le « secteur » doit être traité avec le maximum de précautions.

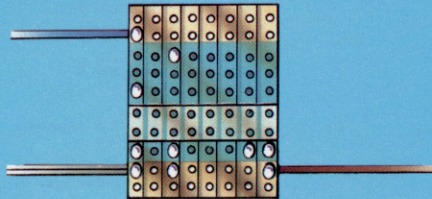
- Déconnectez toutes les sources d'alimentation avant de travailler sur une des pièces du boîtier.
- Vérifiez toutes les connexions et toutes les isolations avant d'appliquer la tension.
- Soyez prudent et évitez de créer des courts-circuits. Souvenez-vous : **LES TENSIONS SECTEUR PEUVENT TUER!**

## La carte

Coupez la carte selon la forme illustrée afin qu'elle soit logée exactement dans l'un des coins du boîtier. Faites les ruptures de pistes et soudez le relais de la façon illustrée dans le diagramme.

**Vérifiez soigneusement la carte avant de continuer.**

Utilisez le multimètre pour vérifier l'absence de shuntage entre les pistes : une erreur à ce niveau peut être mortelle! Soudez le conducteur brun du secteur et le câble-ruban à deux voies sur la carte. Ouvrez l'une des deux fentes préformées sur le boîtier pour y faire passer les fils, mais faites un nœud avec ceux-ci avant de les passer dans l'ouverture : ce nœud évitera que la carte soit endommagée en tirant accidentellement sur les fils. Soudez un conducteur de secteur isolé de courte longueur au boîtier et connectez-le aux bornes à vis de la prise. Connectez les conducteurs bleu et jaune/vert du secteur aux bornes neutres et à la terre respectivement.



## Programme de test

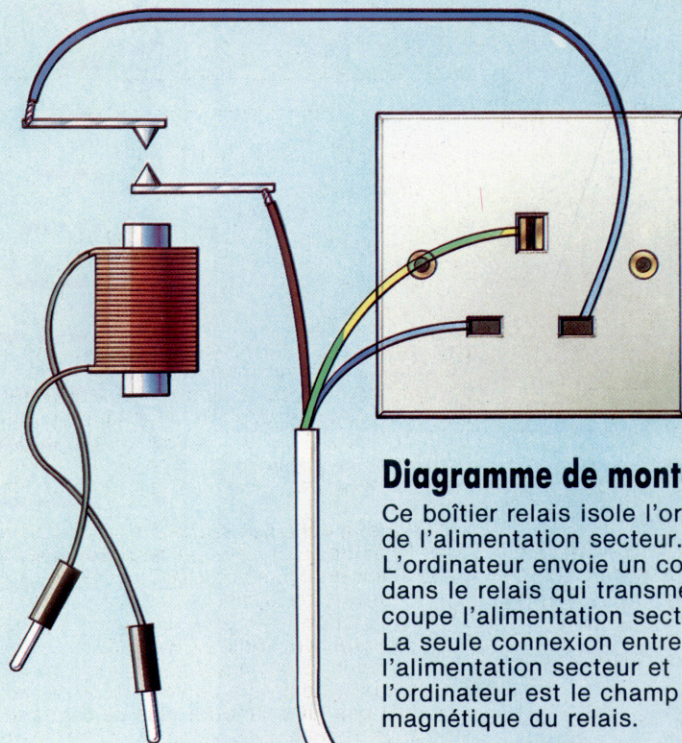
Après avoir construit le boîtier relais et vérifié toutes les connexions, nous pouvons tester son fonctionnement en écrivant un court programme qui commande un dispositif alimenté par le secteur. Une lampe serait parfaitement indiquée. Celle-ci pourrait être branchée à la prise du secteur du boîtier relais, et les fils de signaux connectés aux bornes positives et négatives de la ligne 0 du boîtier de sortie basse tension. Les fils de signaux peuvent être connectés à l'une des bornes sans modifier le fonctionnement du relais. Le conducteur de secteur du boîtier relais est alors branché dans une prise murale.

Après avoir fait les connexions, tapez le court programme suivant; il allume la lampe pendant cinq secondes, puis l'éteint.

```
10 REM TEST RELAIS SECTEUR
20 DDR=&F6E2:DATREG=&F6E0
30 ?DDR=255: REM TOUTES SORTIES
40 ?DATREG=1: REM ALLUME LAMPE
50 TEMPS=0: REM REGLE MINUTERIE
60 REPEAT
70 UNTIL TEMPS > 500
80 ?DATREG=0: REM ETEINT LUMIERE
```

```
10 REM TEST RELAIS SECTEUR CBM 64
20 DDR=56579: DATREG=56577
30 POKE DDR,255: REM TOUTES SORTIES
40 POKE DATREG,1: REM ALLUME LAMPE
50 T=TI: REM REGLE MINUTERIE
60 IF (TI-T) 300 THEN 60
70 POKE DATREG,0: REM ETEINT LAMPE
```

Si, après avoir exécuté le programme, la lampe ne s'allume pas, débranchez le boîtier relais du secteur avant de tester les connexions.

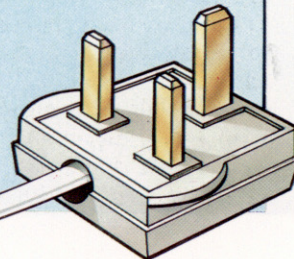


## Diagramme de montage

Ce boîtier relais isole l'ordinateur de l'alimentation secteur.

L'ordinateur envoie un courant dans le relais qui transmet ou coupe l'alimentation secteur.

La seule connexion entre l'alimentation secteur et l'ordinateur est le champ magnétique du relais.





# Projet de code en morse

```

10 REM*****
11 REM*      EXERCICE MORSE C64      *
12 REM*      BRANCHE UNE LAMPE DANS *
13 REM*      RELAIS SECTEUR         *
14 REM*      ENTRE TOUTE CHAINE DE LETTRES *
15 REM*      ET ELLE SERA TRADUITE  *
16 REM*      EN MORSE SOUS FORME    *
17 REM*      DE CLIGNOTEMENTS ET DE BIP SONORES *
20 REM*****
100 GOSUB 2000:      REM INIT
150 FOR L=0 TO 1 STEP 0
200 PRINT "ENTRE VOTRE MESSAGE"
220 PRINT "TAPE 'BYE' POUR SORTIR"
240 INPUT"MESSAGE ";MS$
300 ML=LEN(MS$):M$=""
320 FOR K=1 TO ML
330 C$=MID$(MS$,K,1)
340 IF C$="A"ANDC$="Z" THEN M$=M$+C$
350 IF C$=" " THEN M$=M$+C$
360 NEXT K:IF M$="" THEN NEXT L
400 ML=LEN(M$)
420 FOR K=1 TO ML
440 CH$=MID$(M$,K,1):CH=ASC(CH$)-64
450 IF CH=-32 THEN FORPP=1TO6+DE:NEXTPP
460 IF CH -32 THEN GOSUB 3000
480 FOR PP=1 TO 3+DE:NEXT PP
500 NEXT K
550 IF M$="BYE" THEN L=1
600 NEXT L
900 END
2000 REM*****INIT*****
2100 DIM M$(26)
2110 DDR=56579:DATRQ=56577:POKE DDR,255
2120 DE=25:RX=2+DE
2130 V=54296:LF=54272:HF=54273:W=54275
2140 A=54277:S=54278
2150 FOR K=LF TO LF+24:POKEK,0:NEXT K
2160 POKEA,24:POKES,129:POKEV,15
2200 DATA "-","-.-","-.-.-","-.-.-.-","-.-.-.-.-"
2220 DATA "-.-.-.-.-","-.-.-.-.-.-","-.-.-.-.-.-.-"
2240 DATA "-.-.-.-.-.-.-.-","-.-.-.-.-.-.-.-.-"
2260 DATA "-.-.-.-.-.-.-.-.-.-","-.-.-.-.-.-.-.-.-.-.-"
2280 DATA "-.-.-.-.-.-.-.-.-.-.-.-","-.-.-.-.-.-.-.-.-.-.-.-.-"
2300 DATA "-.-.-.-.-.-.-.-.-.-.-.-.-.-","-.-.-.-.-.-.-.-.-.-.-.-.-.-.-"
2400 FOR K=1 TO 26:READ M$(K):NEXT K
2990 RETURN
3000 ***CLIGNOTEMENT ET BIP SONORE***
3050 PRINT CH$,M$(CH)
3100 N=LEN(M$(CH))
3200 FOR C=1 TO N
3220 D=DE-(ASC(MID$(M$(CH),C,1))-46)*RX
3240 POKE DATRG,I      :REM CLIGNOTEMENT
3250 POKE LF,172:POKE HF,57:REM BIP
3260 POKE W,33:FOR PP=1 TO D:NEXT PP
3270 POKE W,32
3280 POKE LF,0:POKE HF,0 :REM FIN BIP
3290 POKE DATRG,0      :REM CLIGNOTEMENT
3300 FOR PP=1 TO DE:NEXT PP
3320 NEXT C
3490 RETURN

```

```

10 REM*****
11 REM*      EXERCICE MORSE BBC      *
12 REM*      BRANCHE UNE LAMPE DANS *
13 REM*      RELAIS SECTEUR         *
14 REM*      ENTRE TOUTE CHAINE DE LETTRES *
15 REM*      ET ELLE SERA TRADUITE  *
16 REM*      EN MORSE SOUS FORME    *
17 REM*      DE CLIGNOTEMENTS ET DE BIP SONORES *
20 REM*****
100 PROCinitialise
120 ALLOVER=FALSE
150 REPEAT
200 PRINT"ENTRE MESSAGE"
220 PRINT "TAPE 'BYE' POUR SORTIR"
240 INPUT"MESSAGE",MS$
300 ML=LEN(MS$):M$=""
320 FOR K=1 TO ML
330 C$=MID$(MS$,K,1)
340 IF C$="A"ANDC$="Z"THEN M$=M$+C$
350 IF C$=" " THEN M$=M$+C$
360 NEXT K:IF M$="" THEN UNTIL ERREUR
400 ML=LEN(M$)
420 FOR K=1 TO ML
440 CH$=MID$(M$,K,1):CH=ASC(CH$)-64
450 IF CH=-32 THEN PROCretard(6+DE*IX)
460 IF CH -32 THEN PROCbipclignotant
480 PROCretard(3+DE)
500 NEXT K
550 IF M$="BYE" THEN COUVERTURE=VRAI
600 UNTIL COUVERTURE
900 END
2000 REM*****INIT*****
2050 DEFPROCinitialise
2100 DIM M$(26)
2110 DDR=&FE62:DATRQ=&FE60:?DDR=255
2120 DE=3:RX=2+DE:IX=30
2200 DATA "-","-.-","-.-.-","-.-.-.-","-.-.-.-.-"
2220 DATA "-.-.-.-.-","-.-.-.-.-.-","-.-.-.-.-.-.-"
2240 DATA "-.-.-.-.-.-.-.-","-.-.-.-.-.-.-.-.-"
2260 DATA "-.-.-.-.-.-.-.-.-.-","-.-.-.-.-.-.-.-.-.-.-"
2280 DATA "-.-.-.-.-.-.-.-.-.-.-.-","-.-.-.-.-.-.-.-.-.-.-.-.-"
2300 DATA "-.-.-.-.-.-.-.-.-.-.-.-.-.-","-.-.-.-.-.-.-.-.-.-.-.-.-.-.-"
2400 FOR K=1 TO 26:READ M$(K):NEXT K
2990 ENDPROC
3000 REM***CLIGNOTEMENT ET BIP**
3020 DEFPROCbipclignotement
3050 PRINT CH$,M$(CH)
3100 N=LEN(M$(CH))
3200 FOR C=1 TO N
3220 D=DE-(ASC(MID$(M$(CH),C,1))-46)*RX
3240 ?DATRG=1      :REM CLIGNOTEMENT
3260 SOUND I,-15,200,D :REM BIP
3270 PROCretard(IX*D)
3290 ?DATRG=0      :REM FIN CLIGNOTEMENT
3300 PROCretard(DE*IX)
3320 NEXT C
3490 ENDPROC
4000 REM*****RETARD*****
4100 DEFPROC-retard(temps)
4200 FOR DD=1 TO temps:NEXT DD
4490 ENDPROC

```

### Le morse

A	.-
B	-...-
C	-.-.-.
D	-.-.-
E	..
F	..-.-
G	-.-.-
H	....
I	..
J	.-.-.-
K	-.-.-
L	.-.-.
M	---
N	---.
O	---
P	.-.-.-
Q	---.-
R	.-.-.
S	...-
T	---
U	...-
V	...--
W	---.-
X	---..
Y	---.-
Z	---..
,	...--
.	...--





# Alerte rouge

**Missile Command est sans doute l'un des jeux les plus féroces jamais créés. Il vous faut défendre la civilisation face à d'innombrables vagues de missiles nucléaires.**

Pour connaître le succès, un jeu d'arcades doit d'abord retenir l'attention. C'est l'affaire du générique qui réapparaît chaque fois que la partie est terminée, en attendant l'arrivée d'un autre joueur. Ensuite, il doit se révéler immédiatement passionnant. Il suffit qu'une de ces deux conditions ne soit pas remplie pour que le propriétaire de la salle de jeux installe vite une nouvelle machine. Missile Command, s'il ne fait plus fureur comme autrefois, reste une grande réussite. Le jeu est disponible sur cartouche pour les ordinateurs Atari.

Le scénario est à la fois simple et subtil. Vous commandez une station antimissiles au cours d'une guerre nucléaire et vous devez sauver six grandes villes de la destruction en faisant exploser vos bombes à l'arrivée des projectiles nucléaires ennemis. Ce jeu, du genre « tuez-les tous ! », assouvira, sans aucun doute, vos désirs profonds, où mégalomanie et parfaite galanterie se côtoient.

Au cours du jeu, l'écran affiche une vue générale de la scène : les six villes entourent une structure en forme de pyramide, qui n'est autre que la base antimissile, prête à frapper les engins de mort de l'ennemi. Ceux-ci apparaissent en haut de l'écran. À l'aide d'un manche à balai, vous déplacez une croix que vous mettez sur le chemin d'une des ogives nucléaires. Une pression sur le bouton de mise à feu provoque le lancement d'un missile antimissile, qui explose à l'endroit où se tient la croix, ce qui détruit tous les engins adverses situés dans la limite de son rayon d'action.

Certains d'entre eux, toutefois, sont équipés d'ogives à têtes multiples ; elles se fragmentent en plusieurs éléments, dont chacun peut anéantir une ville entière. Complication supplémentaire, des satellites et des avions volant à basse altitude font leur apparition, et eux aussi lancent des missiles ! Pour gagner des points, la straté-

gie est simple : exterminer tout ce qui bouge. À la fin de chaque vague d'assaut, un pictogramme indique quelles sont les cités que vous avez défendues victorieusement et précise combien d'antimissiles il vous reste.

Naturellement, plus on progresse, plus l'ennemi est rapide et puissant. Il devient vite indispensable de mettre au point une stratégie, plutôt que de chercher à détruire tel ou tel missile. Par exemple, il peut être intéressant de faire exploser à la file toute une série d'armes défensives, qui parviendront, peut-être, à balayer une vague entière.

Comme si tout cela ne suffisait pas, voici des ogives parachutées ! Il est très difficile d'en venir à bout. Votre antimissile doit exploser très exactement au-dessus d'elles, sinon il sera balayé.

Tout irait à peu près bien si vous aviez un stock d'armes illimité, mais ce n'est pas le cas (trente seulement au premier niveau). Si vous gaspillez vos réserves, vous n'aurez qu'à regarder, impuissant, les fusées de l'ennemi rayer de la carte les villes et la base elle-même.

Chaque niveau (qui se différencie des autres par la couleur du fond et de l'avant-plan) comporte deux vagues d'assaut ; après quoi, le score est calculé. Comme dans bien des jeux Atari, on peut passer librement de l'un à l'autre. Le jeu prend fin lorsque toutes les villes ont été détruites. Comme on le voit, c'est une conclusion assez pessimiste que renforce encore l'image d'une explosion apocalyptique, sur laquelle se surimposent les mots THE END.

C'est un jeu disponible sur cartouche pour tous les ordinateurs Atari. Il est vendu avec une grande brochure en couleurs, très supérieure à tout ce qui se fait d'habitude en ce domaine, et qui donne de nombreux détails : comment s'y prendre, comment gagner des points, quels sont les meilleurs moyens de procéder, le tout explicité par des illustrations.

**Missile Command** : pour tous les ordinateurs Atari.

**Éditeurs** : Atari Corporation.

**Auteur** : Atari.

**Manche à balai** : indispensable.

**Format** : Cartouche.



# Trois frères

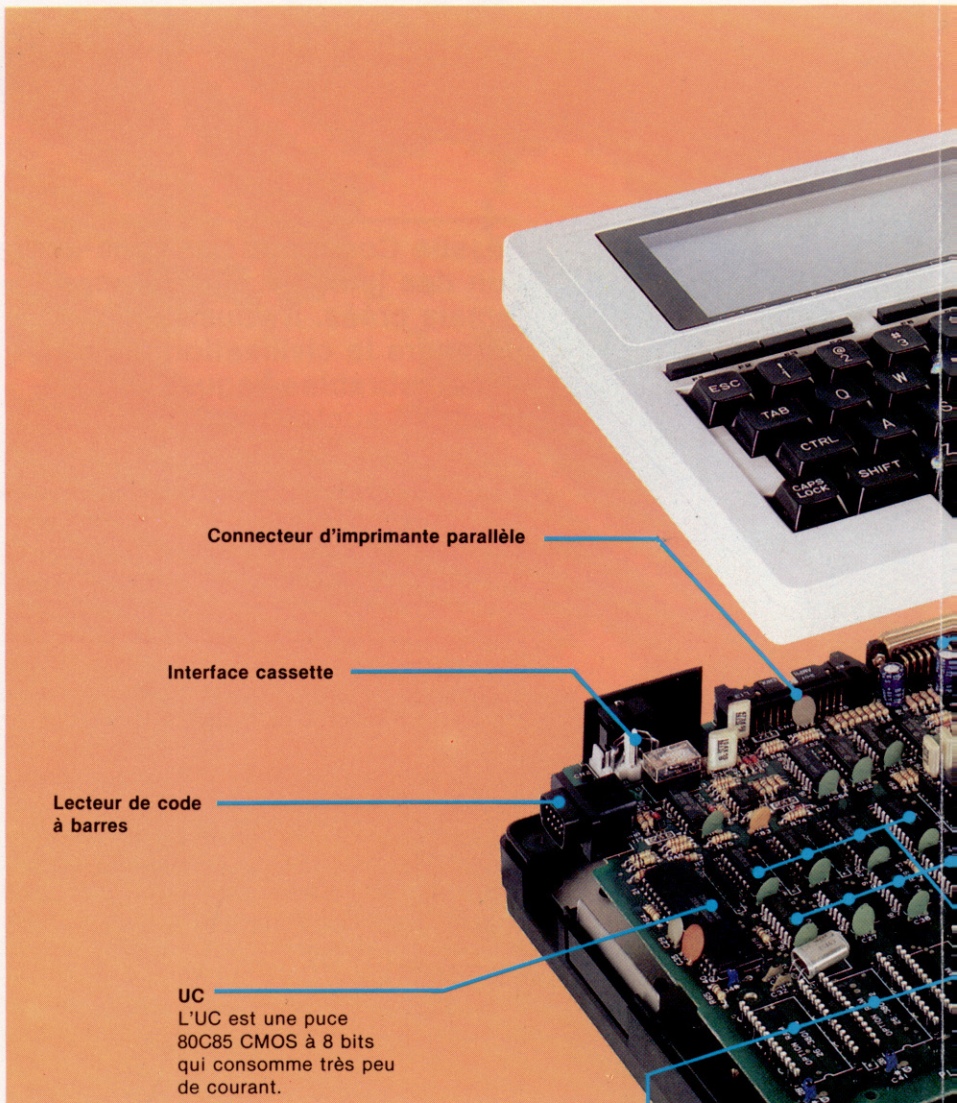
**Les fabricants se lancent de plus en plus sur le marché des ordinateurs portables. Nous comparons le Tandy Modèle 100 avec deux de ses concurrents.**

Deux mêmes produits différant seulement par leur forme, cela existe déjà depuis longtemps dans le domaine de l'électronique grand public — téléviseurs ou chaînes hi-fi. La même technique est maintenant utilisée en informatique. Trois ordinateurs portables — le Tandy 100, le NEC PC-8201-A et l'Olivetti M-10 — proviennent d'un tel accord. Les trois machines sont fabriquées par la société japonaise Kyocera. Tandy, NEC et Olivetti font la finition et assurent la vente sous leurs marques respectives. Nous allons examiner le Tandy Modèle 100 et souligner les différences existant entre cette machine et ses rivaux.

Le Modèle 100 pèse moins de 1,800 kg. Il dispose d'un clavier AZERTY complet, d'un logiciel intégré en ROM et d'un écran à cristaux liquides alimenté par piles. Il peut fonctionner entièrement avec des piles, et le contenu de la RAM n'est pas perdu lors de l'arrêt de la machine. Les fichiers peuvent être stockés en RAM et appelés directement, comme si la mémoire était une cassette ou une disquette. Le Modèle 100 peut aussi être connecté à une unité à cassette ou à un lecteur de disquettes pour y stocker des données, mais la mémoire permanente est plus pratique pour stocker des données importantes pendant le travail.

L'écran à cristaux liquides offre 8 lignes de 40 caractères; il permet de mixer du texte et des graphiques. L'affichage est formé par 15 360 points; chacun peut être adressé individuellement. Les caractères sont formés dans une matrice de 6 par 8, et il est possible d'afficher des caractères minuscules et majuscules. Le Modèle 100 offre un jeu complet de caractères internationaux, ainsi qu'un jeu spécial de caractères graphiques, contrairement au NEC qui n'a que trois caractères graphiques. Les modèles NEC et Tandy possèdent un écran disposé à plat sur le boîtier; l'Olivetti M-10 dispose d'un écran mobile. Le NEC et le Tandy comportent des commandes de contraste pour améliorer la clarté de l'écran.

Le clavier du Tandy comporte des touches spéciales pour appeler les graphiques intégrés ou pour transformer plusieurs touches alphabétiques en un clavier numérique. Les trois machines possèdent des touches de commande du curseur, mais leur position varie. Le Tandy et l'Olivetti ont quatre petites touches adjacentes situées au-dessus, à droite, du clavier régulier, et le NEC PC-8201-A a un clavier curseur où les quatre touches forment un carré.



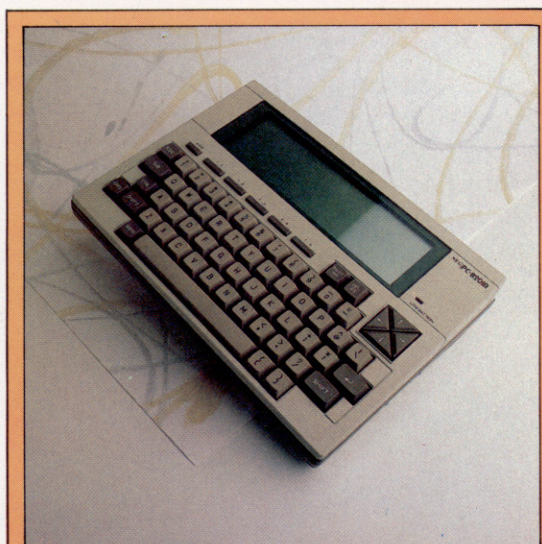
Connecteur d'imprimante parallèle

Interface cassette

Lecteur de code à barres

UC

L'UC est une puce 80C85 CMOS à 8 bits qui consomme très peu de courant.



## NEC PC-8201-A

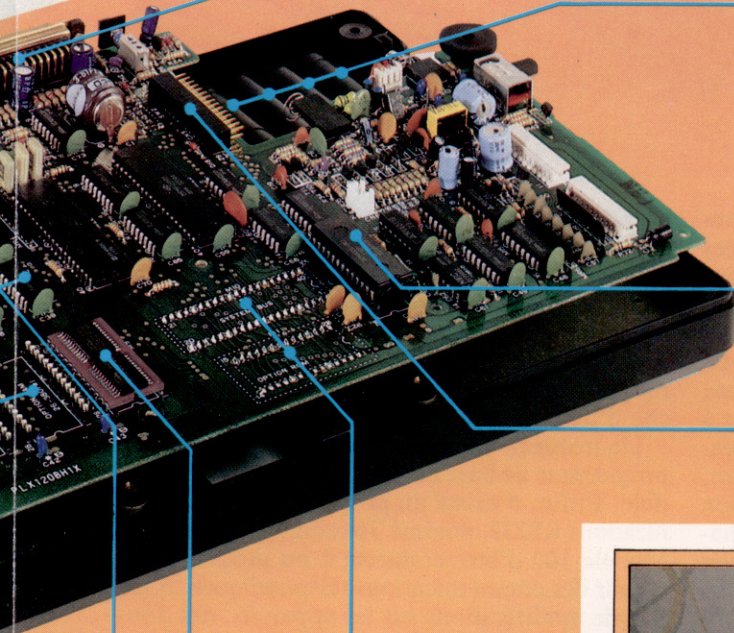
Bien que le NEC PC-8201-A ait exactement les mêmes dimensions que ses concurrents, il a un clavier moins complet. Les touches de commande du curseur ont été placées dans un petit ensemble, les touches de fonction sont passées de huit à cinq, et la disposition du clavier est légèrement différente. Le NEC n'offre en ROM que trois des programmes standard.

## RAM d'extension

Le Tandy 100 peut être étendu de façon interne à une RAM de 32 K en installant ici des puces RAM additionnelles.



**Connecteur de modem**  
Port de communications série RS232 standard. Le logiciel de télécommunication est intégré dans la machine.



**Alimentation**  
Le Tandy 100 peut fonctionner jusqu'à vingt heures avec quatre piles alcalines « AA ». La mémoire interne est conservée jusqu'à trente jours par des piles rechargeables au nickel-cadmium.

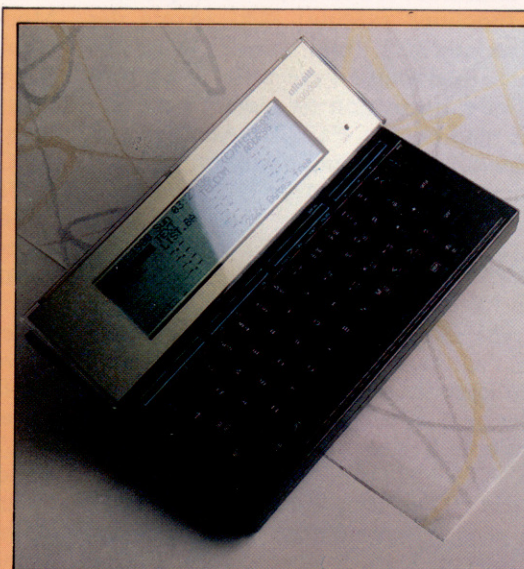
**ROM standard**  
Cette puce renferme le BASIC Microsoft et les logiciels intégrés.

**Connecteur ECL**  
Un câble-ruban relie ici l'écran à cristaux liquides à la carte système.

**Bus système et connecteurs ROM**  
Ces connecteurs vides sont destinés à une extension future de la ROM système et à une commande d'entrée/sortie.

**RAM 8 K standard**

**Clavier**  
Ces puces contrôlent l'entrée et la sortie du clavier et renferment les jeux de caractères.



#### Olivetti M-10

La version Olivetti offre une intéressante caractéristique : l'écran à cristaux liquides peut être incliné à environ 40° — ce qui facilite la lecture de l'affichage. Il possède le même clavier que le Tandy 100 et offre les cinq programmes standard logés en ROM.

## TANDY MODÈLE 100

### PRIX

★ ★ ★

### DIMENSIONS

300 × 215 × 50 mm.

### UC

GMOS 80C85 à 8 bits, 2,4 MHz.

### MÉMOIRE

RAM 8 K ou 24 K, extension par modules de 8 K jusqu'à 32 K; ROM de 32 K incluant les programmes et le BASIC Microsoft.

### ÉCRAN

ECL de 40 colonnes par 8 lignes; graphiques adressables sur 240 par 64 points; caractères ASCII et internationaux, 39 caractères graphiques.

### INTERFACES

Imprimante parallèle, cassette, RS232, port série, lecteur de code à barres, bus système.

### LANGAGE DISPONIBLE

BASIC Microsoft.

### CLAVIER

De machine à écrire standard à 56 touches, clavier numérique intégré; 8 touches de fonction programmables; 4 touches de commande et 4 touches de commande du curseur.

### DOCUMENTATION

Guide de référence rapide de BASIC; manuel d'utilisation détaillé.

### FORCES

Le Tandy offre la plupart des fonctions nécessaires pour un travail « sérieux ». La mémoire permanente et l'alimentation par piles en font une machine vraiment portable.

### FAIBLESSES

Les restrictions au niveau de la RAM limitent le champ d'utilisation de cette machine à des applications d'ordinateur portable. Elle ne peut prétendre remplir les fonctions d'un ordinateur de table.

### DIFFÉRENCES

Olivetti M-10 :  
Clavier à 57 touches;  
47 caractères graphiques,  
écran inclinable; un manuel de l'utilisateur.

### NEC PC-8201-A :

57 touches; 5 touches de fonction; RAM de 16 K avec extension à 96 K; uniquement 3 caractères graphiques.



### Les « puces du vieux »

Même si les Tandy Modèle 100, Olivetti M10 et NEC PC 820/A sont distribués par trois marques différentes, l'essentiel de leur fabrication vient de la même origine, la société japonaise Kyocera. Certes, il existe quelques différences. Mais un œil averti reconnaîtra sans effort l'héritage commun. (Cl. Ian McKinnell.)



Les machines possèdent également des touches de fonction programmables, qui sont utilisées avec le logiciel intégré pour gérer les fonctions de gestion de fichiers et tout mouvement à l'intérieur des programmes stockés en ROM et entre eux. Soulignons de nouvelles différences. Le Tandy Modèle 100 a huit touches de fonction, plus quatre touches additionnelles qui servent à effectuer des tâches internes : PASTE sert à déplacer des données d'un programme à un autre; LABEL attribue des noms aux touches de fonction afin que l'utilisateur se souvienne de ce que chacune des fonctions exécute; PRINT envoie des fichiers directement à l'imprimante; la touche BREAK, enfin, interrompt l'exécution d'un programme. Cette disposition est répétée sur l'Olivetti M-10, mais le NEC PC-8201-A a cinq touches de fonction programmables pour un total de dix fonctions, et une touche PAUSE.

## Capacités mémoire

Le Modèle 100 et le M-10 sont fournis avec 8 ou 24 K de RAM, et cette mémoire peut être portée à 32 K avec l'addition d'un module RAM interne. Le NEC est légèrement différent; il est livré avec 16 K, mais peut être étendu à 64 K de façon interne, ou à 96 K si le port d'extension intégré est utilisé.

Le Modèle 100 est livré avec le BASIC Microsoft et un petit système d'exploitation qui gère le logiciel interne. Lors de la mise sous tension, les fichiers stockés en mémoire sont affichés avec les titres des programmes internes fournis.

Les programmes fournis sont Text, un programme de traitement de texte qui est pratique pour composer des notes internes, des lettres ou de courts rapports; particulièrement indiqué pour

prendre des notes, il devrait séduire les journalistes, les étudiants et les hommes d'affaires. Schedule est un petit programme de base de données, spécialement conçu pour vous aider à vous souvenir de vos rendez-vous, de vos dépenses, des « choses à faire », etc. Une fonction de recherche intégrée facilite et accélère la recherche d'information.

Un troisième programme, Address, est un petit programme de base de données similaire à Schedule. Enfin, il existe un programme de communications RS232, nommé Telecom, qui permet au Modèle 100 d'être connecté à un modem pour établir des communications téléphoniques. Avec peu de modifications sur les appareils, des données peuvent être envoyées à des ordinateurs éloignés ou être reçues de ces mêmes ordinateurs. Le NEC PC-8201-A n'est livré qu'avec BASIC, Text et Telecom.

Les trois machines sont bien équipées en interfaces, et chacune possède un port de communications RS232, un port d'imprimante parallèle, une interface de cassette et une prise pour un lecteur de code à barres. Les modèles Tandy et Olivetti comportent un bus système, tandis que le NEC comprend deux ports série supplémentaires en plus de sa liste d'interfaces.

L'utilisation d'une seule machine de base, avec de légères différences entre les trois modèles, a permis aux fabricants de proposer des produits de haute qualité sans avoir à supporter le coût complet de la mise au point. Astucieuse démarche commerciale!

Mais le plus intéressant n'est-il pas que le client potentiel soit rapidement attiré par le prix relativement peu élevé (autour de 5 000 francs pour les versions de base) de ces matériels, compte tenu de leurs capacités de travail non négligeables.

# Faites la logomotive (!)

Dans cet article LOGO, nous développons un jeu très simple, au cours duquel la tortue se perd dans l'espace. A cette fin, nous étudierons d'abord diverses méthodes d'entrées/sorties.

Dans notre jeu de la Tortue spatiale, la tortue erre au plus profond de l'espace, très loin de sa base, qu'elle doit pourtant rejoindre. Le jeu suppose l'affichage de messages. La commande appropriée est évidemment AFFICHER. Une fois le message affiché, le curseur passe au début de la ligne suivante.

Pour afficher un seul mot, AFFICHER est suivi du mot lui-même. Ainsi, AFFICHER «BONJOUR» affiche le mot «BONJOUR». AFFICHER «» affiche une chaîne de caractères vierges, une ligne vide. Si l'on doit afficher plus d'un mot, le message doit figurer entre crochets pour indiquer qu'il s'agit d'une suite de chaînes de caractères :

```
AFFICHER [LE TEMPS ALLOUÉ EST FINI]
```

AFFICHER permet également d'afficher le contenu d'une variable, aussi AFFICHER:SCORE prend et affiche la valeur de la variable «SCORE. Messages et valeurs de variables peuvent se combiner dans une même instruction AFFICHER, en faisant figurer le tout entre parenthèses, comme dans l'exemple suivant :

```
(AFFICHER [VOICI VOTRE SCORE] :SCORE)
```

AFFICHER1 est comme AFFICHER, à cette différence près que le curseur ne revient pas à la ligne. Essayez :

```
AFFICHER1 [QUEL EST VOTRE NOM?]
```

## Opérations de sortie

La plupart des commandes LOGO, telles que CACHETORTUE ou AFFICHER, ont un effet sur la tortue. D'autres, en revanche, ont pour seul effet de produire une valeur, comme EXCOORD par exemple. Cette valeur est souvent utilisée en entrée avec une commande; aussi, taper :

```
AFFICHER XCOORD
```

fera que XCOORD transmettra à AFFICHER la valeur de l'abscisse de la position courante, qui sera donc affichée. Si la valeur courante de XCOORD est 20, 20 sera alors affiché. Si l'on tape XCOORD seul, le message RÉSULTAT: 20 s'affiche. Il s'agit en fait, dans ce cas, d'un message d'erreur (les versions LCSI sont nettement moins affinées et affichent alors VOUS NE PRÉCISEZ PAS CE QUE VOUS FAITES DE 20).

Les procédures que nous avons écrites jusqu'à présent sont toutes des commandes. Pour créer des opérations, il nous faut utiliser la primitive RÉSULTAT. Comme simple exemple, voici une procédure qui donne la distance de la tortue à l'ori-

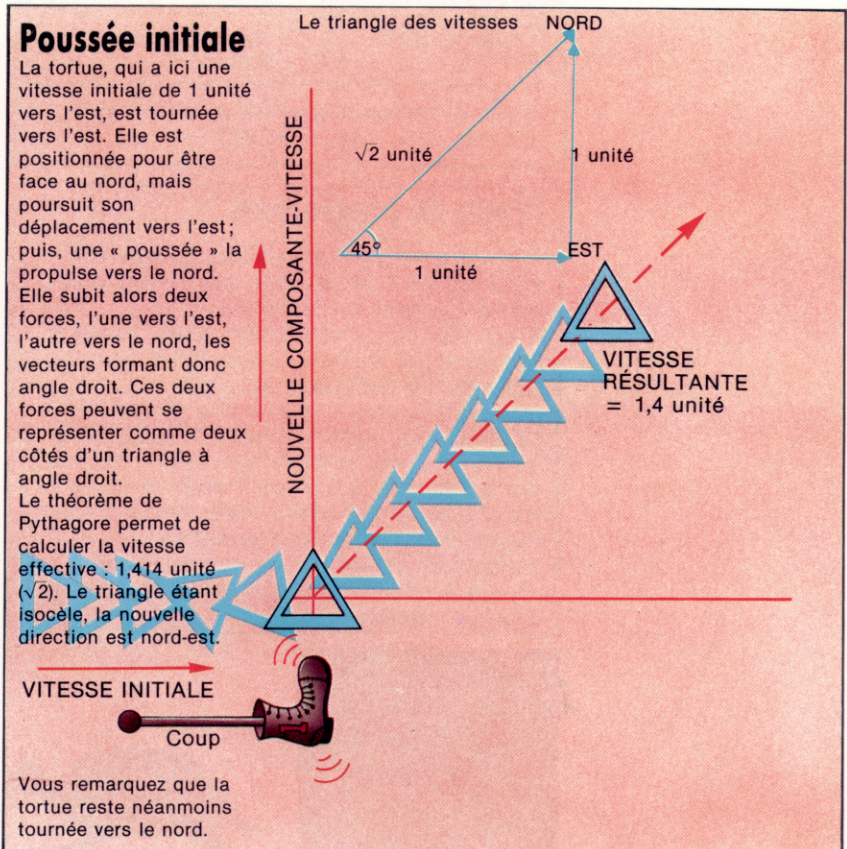
gine. Cette procédure utilise RACINE-CARRÉE, qui calcule la racine carrée d'un nombre.

```
POUR DISTANCE
  RÉSULTAT RACINE-
  CARRÉE (XCOORD*XCOORD + YCOORD*YCOORD)
FIN
```

Déplacez la tortue en différents points de l'écran et utilisez DISTANCE pour connaître sa position par rapport à l'origine. Par exemple, POSITIONNERXY 30 40 AFFICHER DISTANCE donne 50 en réponse.

Lorsque LOGO rencontre une instruction RÉSULTAT, il s'arrête dans l'exécution de la procédure courante et redonne le contrôle à la procédure d'appel. On peut s'en apercevoir avec la procédure MAX, qui donne le plus grand nombre d'un ensemble de deux nombres :

```
POUR MAX :X :Y
  SI :X > :Y ALORS RÉSULTAT :X
  RÉSULTAT :Y
FIN
```





**Tortue extraterrestre**

Le programme tel qu'il est générera seulement la tortue et sa cible (sa base). Les étoiles et les planètes qui figurent ici ont été ajoutées au moyen de procédures simples de cercles. (Cl. Ian McKinnell.)

AFFICHER MAX 6 2 donne 6 comme résultat. Écrivez une procédure qui donne la valeur absolue d'un nombre, de sorte que AFFICHER ABS 4 et AFFICHER ABS (-4) donnent le même résultat, 4.

Notre jeu vous demandera d'entrer votre nom et de faire RC (Retour-Chariot). Voici une procédure :

```

POUR PRENDRE.NOM
  FENÊTRE.D'ÉCRAN
  AFFICHER1 [QUEL EST VOTRE NOM?]
  FAIRE «PREMIÈRE DEMANDE DE NOM
  AFFICHER «BONJOUR :NOM)
  FIN
  
```

DEMANDE attend une ligne de saisie terminée d'un RC. Il donne ensuite la ligne comme une liste. PREMIER donne le premier élément d'une liste. Essayez la procédure PRENDRE.NOM et répondez «SAINT» pour le nom. Voyez à présent ce qui arrive avec « Saint Béat » en entrée.

**Nuances LOGO**

LOGO MIT	LOGO LCSI
TRACE	CS
AFFICHE1	TAPE
RC?	TOUCHEP
LIRECARACTÈRE	LC
DEMANDE	RL
BONNEDIRECTION	DR
POSITIONNEXY	DONNEPOS (suivi d'une liste)
PLEINÉCRAN	FS (Non disponible sur le Spectrum)
FENÊTRED'ÉCRAN	SS (Non disponible sur le Spectrum)

Si est de syntaxe légèrement différente avec LOGO LCSI; ainsi : SI DISTANCE < 5 [ARRÊT]

**Projet de programme**

Écrivez un programme d'alunissage. Pour ceux qui ne connaissent pas ce jeu, voici un bref rappel de ses objectifs. Vous êtes aux commandes d'une fusée à une certaine distance de la Lune, et vos réserves de carburant sont limitées. La gravité lunaire exerce une attraction continue sur votre vaisseau et augmente sa vitesse vers le bas d'un certain facteur par seconde. La touche F commande la mise à feu d'une rétrofusée, freinant votre descente mais brûlant beaucoup de carburant. Le but du jeu est de doser l'utilisation de votre rétrofusée pour vous poser en douceur.

**Abréviations**

RÉSULTAT	RS
AFFICHER	AF
LIRECARACTÈRE	LC
DEMANDE	DM
DONNEDIRECTION	DR



Le jeu contrôle les mouvements de la tortue à l'écran au moyen des touches D, G, et P; D pour tourner la tortue, dans le sens des aiguilles d'une montre, de 30°; G la fera tourner d'autant, mais vers la gauche; P est utilisé pour « pousser » la tortue, pour accroître sa vitesse dans son mouvement présent. Nous déplacerons la tortue dans tout l'écran, et elle devra répondre instantanément à ces commandes. Il serait bien utile qu'une primitive LOGO indique la dernière touche frappée. Si c'était le cas, nous pourrions écrire :

```

POUR COMMANDE
  Faire «COM LIREFRAPPE
  SI :COM = «D ALORS DROITE 30
  SI :COM = «G ALORS GAUCHE 30
  SI :COM = «P ALORS POUSSER
  
```

FIN  
Malheureusement, une telle primitive n'existe pas! Nous pouvons cependant l'écrire en tant que procédure :

```

POUR LIREFRAPPE
  SI RC? ALORS RÉSULTAT LIRECARACTÈRE
  RÉSULTAT)
  
```

FIN  
Lorsque l'on frappe une touche, cette dernière est sauvegardée dans le tampon du clavier. LIRECARACTÈRE donne simplement le dernier caractère dans la mémoire tampon. Si ce dernier est vide, LIRECARACTÈRE attend la frappe d'une touche pour en donner communication. RC? est vrai lorsqu'il y a quelque chose au tampon et faux lorsque ce dernier est vide. Aussi, LIREFRAPPE donne le dernier caractère présent au tampon, ou une chaîne vide lorsque le tampon mémoire ne contient pas de caractère.

Notre tortue de l'espace est en réalité très « dynatortue », une tortue dynamique... Elle avance à une certaine vitesse, possède une position et une direction tout comme ses sœurs sur le plancher des vaches. Mais comme elle est là-haut, dans l'espace, elle ne connaît ni gravité ni forces de frottement. La dynatortue est soumise aux lois de Newton. L'illustration le montrera mieux, mais supposons qu'elle traverse l'écran de droite à gauche à une vitesse de 1. Si on appuie sur G, elle tournera et fera face au haut de l'écran, mais son énergie cinétique fera qu'elle se déplacera toujours horizontalement. Si on appuie sur P, elle subira une impulsion; elle sera poussée en avant dans la direction où elle est tournée. Cela aura pour résultat d'imprimer une poussée vers le haut à la vitesse 1, et la dynatortue se déplacera alors diagonalement sur l'écran avec une vitesse de 1,4. Elle vous permet donc d'expérimenter les lois de déplacement d'un corps physique selon Newton.

Dans le programme, la vitesse de dynatortue est fonction de deux paramètres suivant les axes de l'abscisse et de l'ordonnée. Il s'agit des fonctions trigonométriques SIN et COS. Le seul contrôle du jeu se fait par les touches déjà mentionnées. Pour démarrer le jeu, faites DÉPART. Vous disposez d'un temps limite pour atteindre votre but, et le programme tient la comptabilité du meilleur score.



**Correction des exercices****1. Triangles imbriqués**

POUR TRI :LONGUEUR:NIVEAU  
 SI :NIVEAU = 0 ALORS RÉPÈTE 3 [AV :LONGUEUR DR 120]  
 STOP  
 TRI ( :LONGUEUR /2) ( :NIVEAU - 1)  
 AV ( :LONGUEUR /2)  
 TRI ( :LONGUEUR /2) ( :NIVEAU - 1)  
 DR 60  
 TRI ( :LONGUEUR /2) ( :NIVEAU - 1)  
 AV ( :LONGUEUR /2)  
 DR 60  
 TRI ( :LONGUEUR /2) ( :NIVEAU - 1)  
 GH 60  
 AR ( :LONGUEUR /2)  
 GH 60  
 AR ( :LONGUEUR /2)  
 FIN

**2. Flocon carré**

POUR FLOCON 1:LONGUEUR :NIVEAU  
 RÉPÈTE 4[CÔTÉ1 :LONGUEUR :NIVEAU DR 90]  
 FIN

POUR CÔTÉ1 :LONGUEUR :NIVEAU  
 SI :NIVEAU = 0 ALORS AV :LONGUEUR STOP  
 CÔTÉ1 ( :LONGUEUR /3) ( :NIVEAU - 1)  
 GH 90  
 CÔTÉ1 ( :LONGUEUR /3) ( :NIVEAU - 1)  
 DR 90  
 CÔTÉ1 ( :LONGUEUR /3) ( :NIVEAU - 1)  
 DR 90  
 CÔTÉ1 ( :LONGUEUR /3) ( :NIVEAU - 1)  
 GH 90  
 CÔTÉ1 ( :LONGUEUR /3) ( :NIVEAU - 1)  
 FIN

**3. Courbe dépourvue de gradient**

POUR W :XPAS :YPAS :NIVEAU  
 WHAUT :XPAS :YPAS :NIVEAU  
 WBAS :XPAS :YPAS :NIVEAU  
 FIN

POUR WHAUT :XPAS :YPAS :NIVEAU  
 SI :NIVEAU = 0 ALORS POSITIONNERXY (XCOOR +  
 :XPAS) (YCOOR + :YPAS) STOP  
 WHAUT ( :XPAS /6) ( :YPAS /2) ( :NIVEAU - 1)  
 WBAS ( :XPAS /6) ( :YPAS /2) ( :NIVEAU - 1)  
 WHAUT ( :XPAS /6) ( :YPAS /2) ( :NIVEAU - 1)  
 WHAUT ( :XPAS /6) ( :YPAS /2) ( :NIVEAU - 1)  
 WBAS ( :XPAS /6) ( :YPAS /2) ( :NIVEAU - 1)  
 WHAUT ( :XPAS /6) ( :YPAS /2) ( :NIVEAU - 1)  
 FIN

POUR WBAS :XPAS :YPAS :NIVEAU  
 SI :NIVEAU = 0 ALORS POSITIONNERXY (XCOOR +  
 :XPAS) (YCOOR + :YPAS) STOP  
 WBAS ( :XPAS /6) ( :YPAS /2) ( :NIVEAU - 1)  
 WHAUT ( :XPAS /6) ( :YPAS /2) ( :NIVEAU - 1)  
 WBAS ( :XPAS /6) ( :YPAS /2) ( :NIVEAU - 1)  
 WBAS ( :XPAS /6) ( :YPAS /2) ( :NIVEAU - 1)  
 WHAUT ( :XPAS /6) ( :YPAS /2) ( :NIVEAU - 1)  
 WBAS ( :XPAS /6) ( :YPAS /2) ( :NIVEAU - 1)  
 FIN

**Programme de la tortue spatiale**

POUR COMMENCER  
 FAIRE «MAX 0  
 FAIRE «LE MEILLEUR»  
 TRACER  
 LE.PLUS.GRAND  
 CIBLE  
 JOUEUR  
 FIN

POUR CIBLE  
 CL POSITIONNERXY 0 5 CB  
 DR 90  
 RÉPÈTE 36[AV 31.4/36 DR 10]  
 CL  
 FIN

POUR JOUER  
 PRENDRE.NOM  
 INIT  
 PILOTER  
 FIN

POUR PRENDRE.NOM  
 FENÊTRE.D'ÉCRAN  
 AFFICHER1 [QUEL EST VOTRE NOM?]  
 FAIRE «PREMIÈRE DEMANDE DE NOM»  
 FIN

POUR INIT  
 FAIRE «SCORE 200  
 POSITIONNERXY 100 100  
 SETH 270  
 FAIRE «XVITESSE 0  
 FAIRE «YVITESSE 0  
 PLEINÉCRAN  
 ST  
 FIN

POUR PILOTER  
 COMMANDE  
 DYNA.DÉPLACEMENT  
 SI DISTANCE < 5 ALORS TERMINÉ STOP  
 FAIRE «SCORE :SCORE - 1  
 SI :SCORE = 0 ALORS LE.TEMPS.EST.  
 ÉCOULÉ  
 STOP  
 PILOTER  
 FIN

POUR COMMANDE  
 FAIRE «COM LIREFRAPPE  
 SI :COM = «D ALORS DROITE 30  
 SI :COM = «G ALORS GAUCHE 30  
 SI :COM = «P ALORS POUSSÉE  
 FIN

POUR LIREFRAPPE  
 SI RC? ALORS RÉSULTAT  
 LIRECARACTÈRE

RÉSULTAT>>  
 FIN

POUR POUSSÉE  
 FAIRE «XVITESSE + 3 \* DANS LE SENS  
 DU SIN  
 FAIRE «YVITESSE + 3 \* DANS LE SENS  
 DU COS

FIN  
 POUR DYNA.DÉPLACEMENT  
 POSITIONNERXY XCOOR + :XVITESSE  
 YCOOR + :YVITESSE  
 FIN

POUR DISTANCE  
 RÉSULTAT MISE  
 (XCOOR \* XCOOR + YCOOR \* YCOOR)  
 FIN

POUR TERMINÉ  
 AFFICHER  
 FENÊTRE.D'ÉCRAN  
 (AFFICHER [BIEN VU] :NOM)  
 (AFFICHER [VOICI VOTRE SCORE]  
 :SCORE)  
 BILAN  
 ENCORE  
 FIN

POUR BILAN  
 SI :SCORE > :MAX ALORS FAIRE  
 «MAX :SCORE FAIRE «MEILLEUR :NOM  
 AFFICHER»  
 (AFFICHER [LE MEILLEUR SCORE EST  
 CELUI DE] :MEILLEUR : MAX (POINTS))  
 FIN

POUR ENCORE  
 AFFICHER1 [UNE AUTRE PARTIE?]  
 FAIRE «RÉPONDEZ A LA PREMIÈRE  
 DEMANDE  
 SI :RÉPONSE = «OUI ALORS NOUVEAU.JEU  
 STOP  
 SI :RÉPONSE = «NON ALORS STOP  
 AFFICHER [DÉCIDEZ-VOUS, C'EST OUI OU  
 C'EST NON?]  
 ENCORE  
 FIN

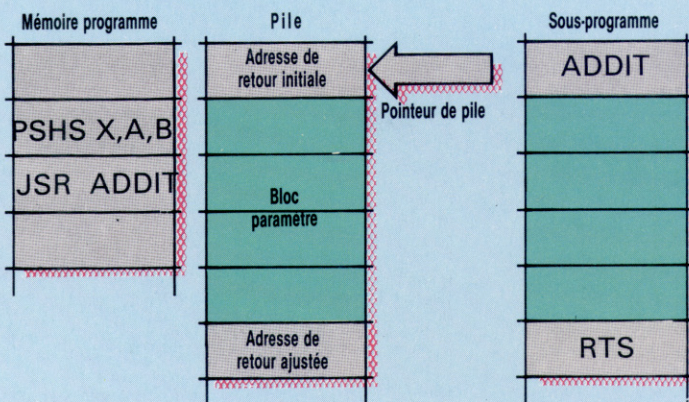
POUR LE.TEMPS.EST.ÉCOULÉ  
 AFFICHER  
 FENÊTRE D'ÉCRAN  
 AFFICHER [LE TEMPS ALLOUÉ EST FINI]  
 ENCORE  
 FIN

POUR NOUVEAU.JEU  
 LE.PLUS.GRAND  
 PRENDRE.NOM  
 INIT  
 PILOTER  
 FIN

# Monter à zéro

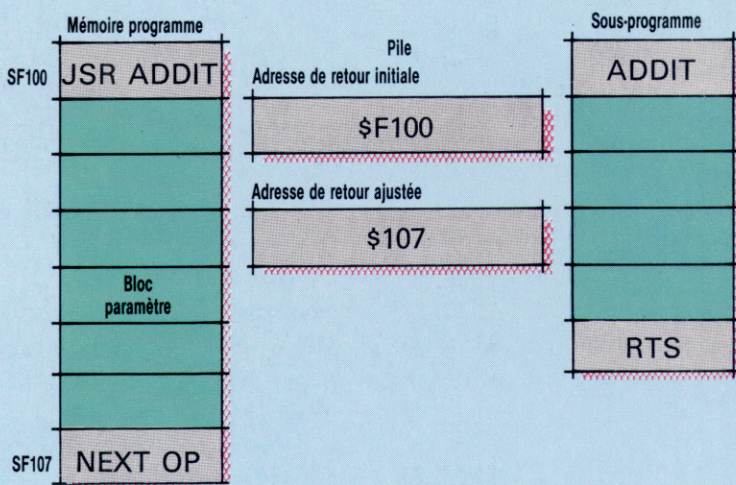
Nous avons à présent un aperçu assez complet des modes d'adressage disponibles sur le processeur 6809. Parmi les variantes, nous allons d'abord examiner les piles en détail.

## Entrée de paramètres



Les paramètres peuvent être passés à un sous-programme en les chargeant dans les registres, puis en les entrant sur pile. Le sous-programme peut les ressortir de la pile, en prenant garde à placer l'adresse de retour JSR au bas de la pile où les paramètres ont été pris. Si ce n'est pas fait, la pile continuera à grandir jusqu'au dépassement.

## Insertion de paramètres



Une méthode plus pratique pour passer les paramètres consiste à les insérer dans le programme directement après l'appel JSR au sous-programme concerné. Le sous-programme peut alors utiliser l'adresse de retour sur la pile comme adresse de base du bloc de paramètres et y accède par adressage indexé. L'adresse de retour doit ensuite être ajustée pour pointer l'instruction de programme suivante, plutôt que le début du bloc de paramètres.

Dans notre cours, nous n'avons jusqu'à présent utilisé les deux registres pointeurs de pile, S et U, que comme registres d'index supplémentaires. L'emploi de la pile matériel pour le stockage des adresses de retour dans les appels de sous-programmes a également été mentionné au passage. Il nous faut maintenant considérer l'architecture d'une pile et son utilisation.

Une pile est un exemple particulier d'un type plus général d'arrangement de données appelé « liste ». Cette notion vous est déjà familière dans la vie de tous les jours, même si vous ignorez tout des langages de traitement de liste, dont la population va croissant : citons LISP et LOGO. Une liste est simplement une suite d'articles de données. Cette suite peut être arrangée dans un ordre déterminé par une quelconque propriété (par exemple une série de nombres dans l'ordre numérique ou une chaîne de caractères dans l'ordre alphabétique) ou bien de manière aléatoire, l'arrangement étant alors déterminé par l'ordre dans lequel les articles de données se sont ajoutés à la liste. Avec toutes ces listes, il est raisonnable d'attacher une signification à l'identité ou à la valeur de l'article *suivant* ou *précédent* dans la liste, et en particulier au premier article (la *tête*) ainsi qu'à son dernier (la *queue*).

Une caractéristique importante d'une liste réside dans le fait qu'il s'agit d'une structure de données dynamique, c'est-à-dire que les articles de données peuvent y être ajoutés ou enlevés, à volonté. Dans une liste générale, les données peuvent être ajoutées ou retirées en n'importe quelle position de la liste. La pile introduit une certaine restriction : une donnée ne peut être ajoutée ou enlevée d'une pile qu'à une extrémité.

Deux situations extrêmes peuvent survenir au cours d'une opération de pile : soit la pile se vide, ce qui ne pose pas de problème si la prochaine opération de pile y ajoute un article, mais pourrait être embarrassant dans le cas contraire ; soit la pile « déborde ».

## Opérations « POP »

Les opérations consistant à ajouter ou à retirer des articles sont désignées respectivement par PUSH et PULL (ou POP). Les deux situations extrêmes que nous venons de mentionner s'appellent « dépassement » (ou « débordement ») et « dépassement négatif ».

Il y a plusieurs manières de mettre en œuvre les piles (en utilisant des tableaux dans un programme BASIC, par exemple), mais la méthode



que nous considérons requiert un bloc de mémoire disponible et un registre, que nous appellerons « pointeur de pile ». Le pointeur est nécessaire pour garder trace de l'emplacement en cours de la tête de liste.

Contrairement à une pile d'assiettes, une pile de mémoire n'est pas accessible à l'inspection, puisqu'il n'y a rien pour distinguer un emplacement mémoire contenant une donnée de pile de l'emplacement suivant, qui peut ne pas faire partie de la pile.

Il faut souligner que, de même qu'une donnée n'est pas vraiment « chargée » depuis la mémoire dans un registre, mais seulement copiée, de même les articles ne sont pas vraiment retirés (PULL) de la pile; c'est seulement le pointeur du sommet de la pile qui est changé.

Le pointeur de pile contient donc l'adresse du sommet de la pile. Il existe deux variantes : le pointeur de pile peut donner soit l'adresse du prochain emplacement libre où la donnée peut être stockée, soit celle du dernier article de données stocké dans la pile. Cette dernière convention est utilisée par le processeur 6809, bien qu'elle ne présente pas d'avantage particulier sur la première méthode.

Une différence d'organisation importante entre une mémoire de pile et une pile d'assiettes dans une armoire apparaît de la façon suivante : dans le premier cas, la croissance se fait vers le bas.

Dans le système 6809, plus on introduit (PUSH) d'articles dans la pile, plus l'adresse du pointeur de pile descend. Une convention veut que l'on dise alors que l'adresse du pointeur de pile « croît vers zéro ».

## Opérations de pile

Les deux opérations de pile du 6809 sont représentées par les instructions PSH (entrée de données sur la pile) et PUL (sortie de la pile). Ces opérations peuvent s'appliquer à l'un ou l'autre des pointeurs S et U, de sorte que nous avons PSHS, PULS, PSHU et PULU. La donnée sur laquelle on opère doit venir d'un registre ou y entrer, bien que plusieurs registres puissent être impliqués dans une instruction.

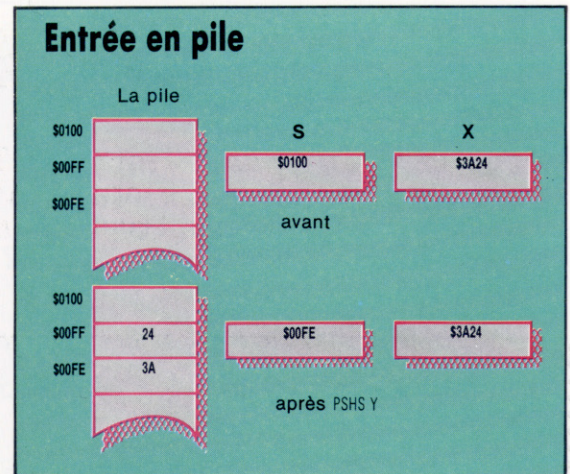
L'instruction PSHS X aura d'abord pour effet de décrémenter S, le pointeur de pile, de 2 (ou d'une unité s'il s'agit d'un registre à 8 bits) pour donner l'adresse du prochain emplacement de pile libre, et ensuite de stocker le contenu de X à cette adresse.

La première figure illustre cette procédure. Notez encore une fois la convention d'adressage hi-lo du 6809 : l'octet hi (\$3A) de X est stocké en \$00FE, position mémoire inférieure à celle de l'octet lo (\$24), qui est stocké en \$00FF. Si vous utilisez un assembleur, ces détails concernant l'incrément ou la décrément du pointeur de pile sont inutiles : l'assembleur se charge de toute la gestion mémoire.

L'instruction PULS X a l'effet opposé : la valeur 16 bits à l'adresse en cours dans S est chargée dans X, et le contenu de S est ensuite incrémenté de 2. La seconde figure illustre ces changements.

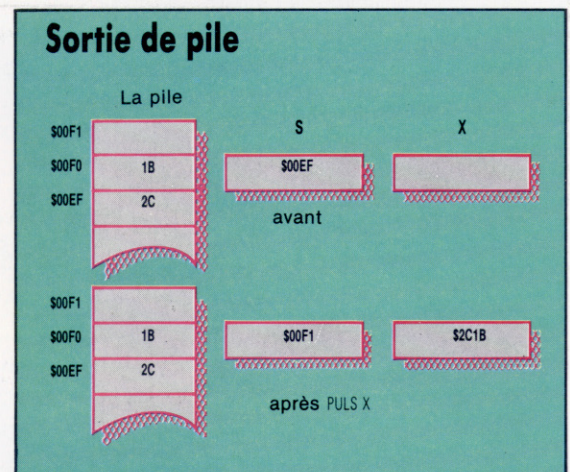
### Entrée en pile

Le pointeur de pile 6809 adresse toujours le « haut » de la pile, c'est-à-dire l'octet le plus récemment inscrit. Lorsque PSHS X est exécuté, S est décrémenté de 2, de sorte qu'il pointe le nouveau sommet de la pile, et le contenu de X (registre à 2 octets) est inscrit à cette adresse en format hi-lo. Notez que la pile « monte à zéro » : le pointeur de pile pointe des emplacements mémoire inférieurs à mesure que la pile croît.



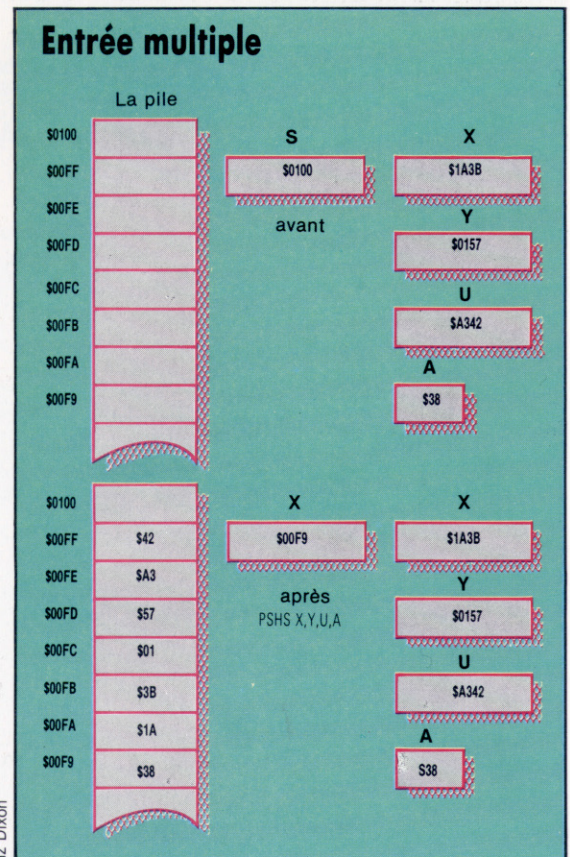
### Sortie simultanée de pile

Lorsque PULS X est exécuté, le contenu des 2 octets à l'adresse en cours du pointeur de pile est copié en X, puis S est incrémenté de 3 pour pointer le nouveau sommet de pile.



### Entrée simultanée de pile

Lorsqu'une opération de pile à registres multiples est exécutée, les registres concernés sont adressés dans un ordre prédéterminé PC,U ou S,X,Y,X,DP,B,A,CC. Lorsque PSHS X,Y,U,A est exécuté, le contenu de U est d'abord entré en pile, suivi de Y,X et A.





### Addition en multiple précision

Voici deux parties de code montrant l'alternative pour effectuer une addition en multiple précision à l'aide de piles. Dans la première partie, les paramètres sont placés après l'appel de sous-programme. Voici un exemple d'appel pour additionner deux nombres à 4 octets en \$100 et \$104, le résultat étant mis en \$108 :

BSR	MPADD		Longueur de chaque nombre (en octets)
FCB	4		Adresse du premier nombre
FDB	\$100		Adresse du second nombre
FDB	\$104		Adresse du résultat
FDB	\$108		Sauvegarde tous les registres. Entre 9 octets sur la pile.
ORG	\$1000		Cette instruction charge dans U l'adresse de retour de PC entrée sur pile par l'instruction BSR
MPADD	PSHS	X,Y,U,A,B,CC	En traitant les données qui viennent après l'appel de sous-programme comme si c'était une pile, on sort les adresses des deux nombres pour les mettre en X et Y, et la longueur en B
	LDU	9,S	
	PULU	X,Y,B	
	LDU	;U	
LOOP	ANDCC	;%11111110	L'adresse du résultat vient en U
	LDA	,X+	
	ADCA	,Y+	Mise à zéro du drapeau de retenue
	STA	,U+	Addition d'un octet à l'autre, avec retenue
	DECB		Stockage du résultat
	BNE	LOOP	Vérifie si l'addition est finie
	LDU	9,S	Sinon, boucle (LOOP)
	LEAU	7,U	Autrement, donne l'adresse de retour et additionne 7 pour sauter les 7 octets de paramètres stockés après l'appel
	STU	9,S	
	PULS	PC,U,X,Y,A,B,CC	Restaure tous les registres à leur état initial, en restaurant le PC au lieu d'utiliser un RTS supplémentaire

Le second exemple effectue la même opération, mais entre les paramètres sur la pile. La séquence d'appel serait la suivante :

LDU	;\$108		Adresse du résultat
LDX	;\$100		Adresse de l'opérande
LDY	;\$104		
LDA	#4		Longueur
PSHS	U,X,Y,A		Entre les paramètres sur pile
BSR	MPADD		
ORG	\$1000		
MPADD	PSHS	X,Y,U,A,B,CC	Sauvegarde tous les registres
	LEAU	11,U	La sauvegarde des registres prend 9 octets, et l'adresse de retour en prend 2 (soit 11 au total). U pointe les paramètres
	PULU	X,Y,B	
LOOP	ANDCC	;%11111110	
	LDA	,X+	
	ADCA	,Y+	Procède comme précédemment
	STA	,U+	
	DECB		
	BNE	LOOP	
	PULS	PC,U,X,Y,A,B,CC	

Il est possible d'entrer ou de sortir en pile plus d'un registre à la fois. Considérons l'instruction :

```
PSHS X,Y,U,A
```

Lorsque plusieurs registres sont ainsi entrés, l'ordre dans lequel les registres sont énumérés est ignoré; ils sont toujours entrés en pile dans cet ordre : PC (registre compteur de programme); U ou S, Y, X, DP (registre de page direct); B, A et CC (registre de code condition). Bien sûr, ils seront toujours sortis de la pile dans l'ordre inverse. La seule vraie contrainte sur les opérations de pile est que ni S ni U ne peuvent être entrés sur leur propre pile.

On se sert de piles en programmation générale pour stocker rapidement et temporairement, mais surtout lorsqu'on traite des interruptions (comme nous le verrons dans la suite de ce cours) et des sous-programmes.

On a déjà vu comment les contenus des registres du compteur de programme sont automatiquement ajoutés dans la pile, quand un sous-programme est appelé, et au contraire retirés du sous-programme (RTS est équivalent à PULS PC). Même une pile, mais particulièrement S, peut également être utilisée pour passer des paramètres à un sous-programme.

La méthode que nous avons utilisée jusqu'ici pour passer des paramètres via les registres (comme dans le programme de table de branchements) présente deux faiblesses majeures. Tout d'abord, il peut y avoir plus de paramètres à passer que de registres; ensuite, cela peut être embarrassant lorsqu'une routine appelée utilise un registre contenant un paramètre qu'il faut conserver. Il existe cependant deux autres techniques courantes pour passer des paramètres :

1. — Les données peuvent être stockées au milieu du programme à l'aide de FCB, FDB ou FCC, immédiatement après l'appel de sous-programme. La valeur du registre compteur de programme entré sur pile par l'instruction JSR donne l'adresse de la première de ces valeurs (puisque PC pointe toujours l'octet suivant l'instruction en cours) et peut servir à les obtenir toutes, avec les décalages adéquats. Le premier programme illustre cette technique. Attention à l'instruction RTS, qui doit passer le contrôle à une instruction réelle, et non à un article de donnée.

2. — Les données peuvent être chargées dans les registres et entrées sur pile avant l'appel de sous-programme, à partir de quoi elles seront entrées dans le sous-programme et utilisées. Attention : ici, à l'instruction RTS, le pointeur de pile accèdera à l'adresse de retour PC préalablement stockée en pile. La seconde partie de code illustre cette technique. C'est généralement une méthode plus utile que la première.

Dans les deux méthodes, le rôle double de S et de U en tant que registres d'index et de pointeurs de pile signifie que les articles de la pile peuvent être référencés par adressage indexé, outre le fait qu'ils sont plus faciles à sortir de la pile. Cela permet de s'assurer plus aisément que les bons articles sont laissés sur la pile pour le retour.



# Sacrée mémoire

**Roland Moreno est l'inventeur français qui a rendu intelligentes les cartes de crédit et les terminaux Télétel. A l'étranger, on envie beaucoup ses découvertes.**

La carte à mémoire a commencé, depuis quelques mois, à faire son apparition en France. Déjà utilisable dans certaines cabines téléphoniques, elle sera bientôt adoptée pour les paiements, retraits d'argent, etc. Une autre carte, placée dans un micro-ordinateur Apple, transforme celui-ci en Minitel intelligent et peut même faire office de serveur Vidéotex. Quel rapport y a-t-il entre ces deux réalisations électroniques? Le rapport, c'est l'homme qui est à l'origine de ces inventions : Roland Moreno.

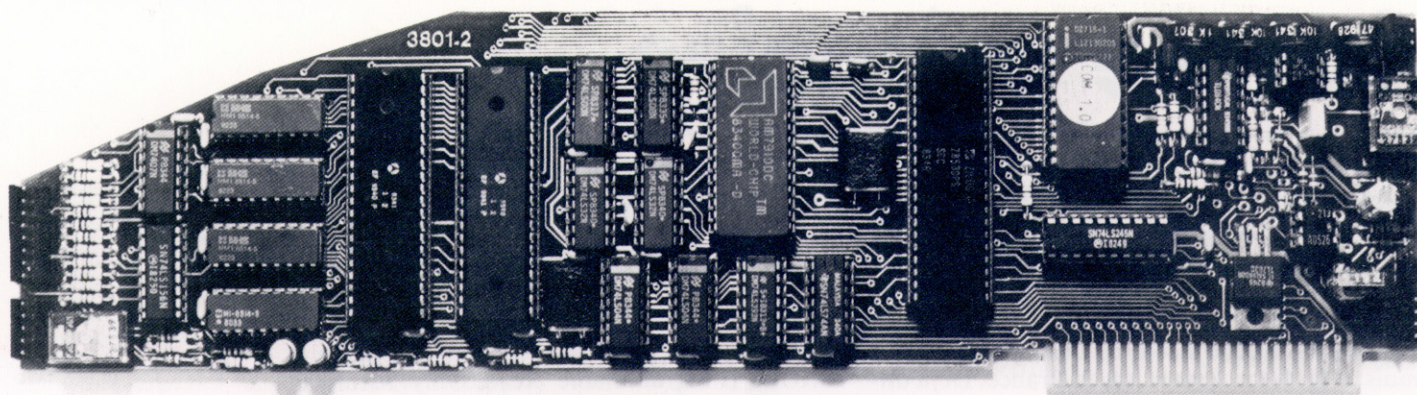
En janvier 1974, cet ingénieur français dépose, au nom de la société Innovatron, les premiers brevets d'une carte dont la lecture et l'écriture sont protégées par un dispositif électronique. C'est une véritable révolution dans le monde des cartes de paiement courantes. Celles-ci, en effet, comportent, pratiquement toutes, une ou plusieurs pistes magnétiques, sur lesquelles sont inscrites un certain nombre d'informations concernant, entre autres, le titulaire de la carte et les types d'utilisation prévus. Or, ces pistes présentent deux inconvénients majeurs : elles sont très vulnérables, comme tout support magnétique, et tout à fait passives vis-à-vis de l'extérieur ; en effet, les pistes magnétiques peuvent être lues et même écrites facilement, autorisant ainsi une utilisation frauduleuse.

La carte à mémoire se distingue de la précédente par le fait que les informations qui y sont contenues sont protégées. Elle combine sur un seul support une mémoire et une intelligence qui protège cette mémoire en contrôlant obligatoire-

ment ses entrées et sorties. Sur la mémoire PROM (Programmable Read Only Memory, « mémoire inscriptible et non effaçable », un certain nombre de données peuvent être inscrites définitivement. Une de ses zones contient le code secret, que l'utilisateur doit taper sur le lecteur de carte pour pouvoir se servir de celle-ci. Alors que, pour la carte magnétique, la comparaison entre les codes se fait dans le lecteur de cartes, dans notre cas, elle se fait à l'intérieur de la carte à mémoire. A chaque tentative d'entrée de code, un compteur est incrémenté en cas d'erreur, donnant droit à trois erreurs au maximum. Au-delà de ce seuil, la carte s'autodétruit. Ainsi, un individu ne connaissant pas le code n'aura pratique-



Cet homme est très demandé, du Japon aux États-Unis. Grâce à son invention — la carte à mémoire —, les cartes de crédit seront pratiquement inviolables. (Cl. Hello Informatique.)



Le secret de la carte à mémoire de Roland Moreno est presque levé par cette photo ! Pour savoir où se situe la véritable révolution, il vaut mieux regarder du côté des applications. Elles sont de plus en plus nombreuses. (Cl. Hello Informatique.)

ment aucune chance de pouvoir utiliser la carte à mémoire; la fraude est ainsi rendue impossible.

La carte à mémoire permet aussi d'empêcher le piratage électronique. Dans ce cas, elle donne l'accès à des systèmes informatiques. Par exemple, une petite entreprise n'ayant pas les moyens de se payer un logiciel de CAO (conception assistée par ordinateur) peut le louer. La carte à mémoire lui donnera accès à ce logiciel, et le prix de la location tiendra compte du temps d'utilisation comptabilisé par la carte à mémoire.

C'est ainsi que les micro-ordinateurs s'équiperont peu à peu pour utiliser ce nouveau média. Sur une récente version de l'IBM PC, déjà disponible aux États-Unis, un des lecteurs de disquettes est remplacé par un lecteur de cartes à mémoire permettant de faire fonctionner des logiciels protégés. Encore aux États-Unis, le département de la Défense a acquis la carte à mémoire pour contrôler l'accès aux magasins et restaurants de son personnel.

Bientôt on pourra effectuer la plupart des transactions bancaires grâce à la carte à mémoire, et celle-ci deviendra une véritable monnaie électronique. Si elle est déjà utilisable dans les publiphones, comme la carte CP8 de Bull à Blois, et la télécarte Schlumberger dans le reste de la France, elle permettra aussi d'accéder aux chaînes de télévision payantes, de faire des achats à domicile par l'intermédiaire d'un Minitel, d'effectuer des réservations de transports, spectacles, etc. Enfin, ce « support individuel d'informations » permet de se dispenser des fichiers centraux.

Les applications de la carte à mémoire sont en fin de compte apparues comme tellement ambitieuses que ce dispositif s'est transformé en média, mettant en jeu un grand nombre d'acteurs. Contrairement à un objet qui peut être utilisé seul, telle une calculatrice, la carte à mémoire remet en cause son environnement : elle nécessite des lecteurs, des équipements spéciaux de banques, cabines téléphoniques... Des pesanteurs administratives et sociologiques ont évidemment ralenti le développement de ce projet national.

Nous ne quittons pas le domaine de la télématique et des médias lorsque nous considérons la seconde grande réalisation de Roland Moreno.

Il y a à peine plus d'un an, alors que Minitel commençait à se répandre dans les foyers français, R. Moreno créa Apple-Tell. Cette carte élec-

tronique, placée dans l'un des connecteurs (slot) d'un micro-ordinateur Apple muni du logiciel spécialisé, permet non seulement l'émulation de Minitel, mais dispense aussi l'utilisateur de toutes les tâches requises pour accéder à une banque de données. Ces tâches sont à la fois fastidieuses, car répétitives, et même parfois humiliantes pour l'utilisateur, qui risque de se faire « injurier » en face de ses collègues ou, pis, de ses collaborateurs subalternes, par sa machine lorsqu'il a fait une fausse manœuvre.

Apple-Tell intègre, en effet, un modem permettant d'automatiser intégralement la consultation Vidéotex. De plus, les informations recueillies peuvent alimenter les programmes utilisés quotidiennement sur l'ordinateur. Cette nouvelle possibilité offerte aux utilisateurs de micro-ordinateurs Apple ne va pas dans le sens d'une augmentation de puissance de traitement, mais vers une plus grande capacité d'entrées/sorties, et elle apporte d'autres avantages. Pour Roland Moreno, l'avenir est aux interfaces utilisateurs plus accessibles. C'est dans cette optique qu'il a conçu Apple-Tell.

Créé en tant qu'« émulateur de Minitel » sur micro-ordinateur Apple II, Apple-Tell n'a pas attendu un an avant de trouver des applications diverses auxquelles son créateur lui-même n'avait pas pensé. Il peut assumer des fonctions de terminal automatique de messagerie ou de banque, serveur Vidéotex, convertisseur de standard, etc. Un de ses utilisateurs, par exemple, réalise grâce à ce système un journal automatique : abonné à une banque de données, son Apple II sélectionne automatiquement toutes les références intéressantes sa spécialité; les publications correspondantes sont alors imprimées, de sorte que, chaque matin, notre homme dispose d'un journal original, parfaitement adapté à ses besoins, qui lui évite l'achat et la consultation de toute une série de revues spécialisées.

Aujourd'hui, la société est prête à accueillir aussi bien Apple-Tell que la carte à mémoire. Mais ce n'était pas le cas il y a dix ans, lorsque celle-ci fut inventée. « Il ne faut pas être trop en avance sur son temps », conclut Roland Moreno. Un brevet ne dure que vingt ans, il entre ensuite dans le domaine public, et son inventeur n'en tire plus aucun bénéfice. Une invention ne peut devenir profitable à son auteur que lorsque ses applications sont déjà « dans l'air ».

**Page manquante  
(publicité)**

**Page manquante  
(publicité)**